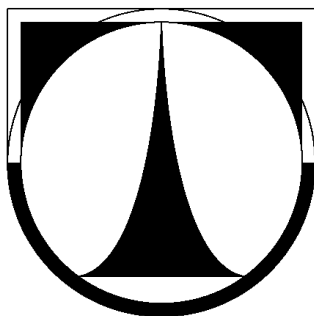


TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta strojní



DISERTAČNÍ PRÁCE

Liberec 2013

Ing. František Koblasa

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta strojní

Studijní program: P2301 STROJNÍ INŽENÝRSTVÍ

Studijní obor: Výrobní systémy a procesy

Zaměření: Automatizace technické přípravy strojírenské výroby

Uplatnění heuristických optimalizačních metod v oblasti rozvrhování strojírenské výroby středních a malých podniků

Application of heuristic optimization methods in the field of small and medium enterprises shop scheduling.

Autor: Ing. František Koblasa

Školitel: doc. Ing. Dr. František Manlig,

Katedra výrobních systémů

Počet stran: 185

Počet příloh: 7

Liberec

2013

Prohlášení

Byl jsem seznámen s tím, že na mou doktorskou disertační práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé disertační práce pro vnitřní potřebu TUL.

Užiji-li disertační práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL, v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložil na vytvoření díla, až do jejich skutečné výše.

Doktorskou disertační práci jsem vypracoval samostatně s použitím uvedených zdrojů a na základě konzultací se školitelem disertační práce a ostatními odborníky v oboru.

Poděkování

Rád bych poděkoval svému školiteli doc. Ing. Dr. Františku Manligovi za trpělivost, odborné vedení a cenné podněty, které pomohly vzniku této disertační práce. Taktéž bych rád poděkoval doc. José António Oliveira Ph.D za odborné vedení v oblasti optimalizace pomocí heuristických metod a cenné rady při tvorbě rozvrhovacího softwaru během i po skončení mé stáže na katedře Výrobních systémů (Department of Production and Systems) na Universidade do Minho. Taktéž děkuji za podporu zaměstnanců Technické univerzity v Liberci zejména pak zaměstnancům Katedry výrobních systémů na Fakultě strojní.

Dále děkuji zaměstnancům TOS VARNSDORF a.s. a MZ Liberec a.s. za poskytnutí informací o jejich výrobním systému a za cenné rady a připomínky při tvorbě reálných modelů vedoucích k ověření metodiky stanovené v této práci.

Velké poděkování patří mým rodičům a přátelům, bez jejichž bezmezná podpora by tato práce nikdy nevznikla.

Anotace

Tato disertační práce se zabývá uplatněním optimalizačních metod při rozvrhování strojírenské výroby malých a středních podniků.

V disertační práci jsou rozebrány současné způsoby plánování a rozvrhování výroby spolu s jejich výhodami a nevýhodami.

Z důvodu obsáhnutí co nejširší oblasti rozvrhování strojírenské výroby jsou v další části analyzované základní typy modelů, které simulují různé trajektorie výrobních zakázek výrobním systémem.

Následně jsou analyzovány současné metody pro tvorbu a optimalizaci výrobních rozvrhů s cílem vybrat vhodné metody aplikovatelné v současné strojírenské výrobě.

Stěženi část práce se zabývá návrhem metodiky využití pokročilých heuristických metod v praxi. Metodika zohledňuje cíle strojírenského podniku a schopnost tyto cíle kvantifikovat. Důležitým prvkem metodiky je analýza současného stavu tzn. současných plánovacích metod uplatněných ve společnosti společně s informacemi o výrobním systému. Na základě těchto informací je vytvořen rozvrhovací model, který slouží k porovnání jednotlivých optimalizačních metod a tvorbě rozvrhu zakázek.

V návaznosti na stanovenou metodiku jsou navrženy vlastní optimalizační algoritmy vycházející z analýzy současných metod. Mezi tyto metody patří modifikovaný konstruktivní algoritmus, lokální prohledávání s jedním prohozením a evoluční algoritmus využívající opravu chromosomu, kontroly klonů a hybridizaci lokálním prohledáváním.

Na teoretických rozvrhovacích modelech jsou provedeny série testů navržených optimalizačních metod s cílem nalezení vhodného nastavení jejich operátorů.

Na základě výsledků těchto testů a požadavků na současný stav rozvrhování ve strojírenské výrobě je navržen princip automatické kaskádové optimalizace, která využívá zmíněné metody tak, aby jedinými vstupujícími parametry byl dostupný čas na optimalizaci a hodnotící kritérium rozvrhu (účelová funkce).

Metodika, včetně navržených metod optimalizace výrobních rozvrhů, je ověřena na modelech reálné výroby pomocí vyvinuté softwarové aplikace.

Klíčová slova: rozvrhování, heuristické algoritmy, metodika využívání heuristických algoritmů

Annotation

This dissertation deals with the application of optimization methods on scheduling problems in the engineering SMEs.

The dissertation analysed the current methods of planning and scheduling along with their advantages and disadvantages.

There are analysed basic types of models that simulate different trajectories of jobs through production system in order to find model which covers widest possible field of shop scheduling problems.

Further, there are analysed the current methods for creating and optimizing production schedules in order to choose appropriate methods applicable in the present mechanical engineering.

The main part of this thesis describes developed methodology for assessing the appropriateness of the use of advanced heuristic methods in practice. The methodology takes into account the objectives of an engineering company and the ability to quantify these objectives. An important element of the analysis is the current state i.e. Current planning methods applied in the company together with information describing the manufacturing system. On the basis of this information is there is created scheduling model, which is used to compare different optimization methods and to create production schedule.

Following the set methodology, there are designed own optimization algorithms based on the analysis of current methods. These methods include modified constructive algorithm, single swap local search and evolutionary algorithm using chromosome repair, clone control and hybridization by local search.

There are performed a series of tests of designed optimization methods in order to find suitable setting of their operators.

Based on the results of these tests and the requirements of the current state of scheduling in mechanical engineering, there is designed cascade principle of automatic optimization that uses before mentioned methods so that the only parameters to set are the time available for optimization and the evaluation schedule (objective function).

Methodology including developed method of schedule optimization is verified on the models of the real production using developed software applications.

Key words: scheduling, heuristic algorithms, methodology of heuristic algorithms application.

Obsah

1	Úvod	21
1.1	Manuální rozvrhování	21
1.2	Rozvrhování pomocí plánovacích tabulí nebo tabulkových kalkulátorů	22
1.3	Rozvrhování v oblasti materiálového plánování	23
1.3.1	Řízení tlakem - ERP/APS a MES systémy	23
1.3.2	Řízení tahem - Štíhlá výroba	24
1.3.3	Řízení pomocí úzkých míst - teorie omezení	24
1.4	Řízení s podporou počítačové simulace	25
1.5	Shrnutí	25
2	Cíle a struktura disertační práce	27
3	Optimalizační problémy a jejich složitost	28
3.1	Časová složitost a třídy optimalizačních algoritmů	29
3.2	Rozvrhovací modely optimalizačních problémů	31
3.2.1	Otevřený problém (Open Shop Scheduling Problem)	33
3.2.2	Proudový problém (Flow Shop Scheduling Problem)	34
3.2.3	Sekvenční problém (Job Shop Scheduling Problem)	36
3.2.4	Smíšený problém (Mixed Shop Scheduling Problem či Group Shop Scheduling Problem)	38
3.2.5	Více komponentní rozvrhovací problém (Multi-component Scheduling Problem)	39
3.3	Dílčí závěr č.1 - Výběr modelu pro co nejširší uplatnění ve strojírenské výrobě	40
3.3.1	Složitost optimalizačních problémů	40
3.3.2	Využití modelu vzhledem k druhu materiálového toku	41
3.3.3	Aplikace modelů v praxi	41
3.3.4	Shrnutí	43
4	Metody pro tvorbu a optimalizaci dílenských rozvrhů	44
4.1	Účelová funkce	44
4.2	Množiny všech řešení a podmnožiny sousedních řešení	48
4.3	Přesné metody řešení	50
4.3.1	Metoda větví a mezí	50
4.3.2	Matematické programování	51
4.4	Přibližné metody řešení	52

4.4.1	Konstruktivní algoritmy	52
4.4.2	Metody založené na lokálním prohledávání	55
4.4.3	Metody založené na evolučních principech	59
4.4.4	Metody založené na umělé inteligenci	65
4.5	Dílčí závěr č.2 – Výběr metod pro tvorbu a optimalizaci dílenských rozvrhů ...	69
5	Metodika využívání heuristických algoritmů v rozvrhování výroby v SME	73
5.1	Problematika zavádění informačních systémů řízení výroby	73
5.2	Definování požadavků na plán – rozvrh	75
5.3	Stanovení měřitelných ukazatelů	78
5.4	Analýza současného stavu	79
5.5	Testování modelu pomocí navržených rozvrhovacích metod	81
5.6	Dílčí závěr č.3 - Shrnutí metodického postupu	84
6	Návrh metod pro tvorbu a optimalizaci dílenských rozvrhů	86
6.1	Konstruktivní algoritmus	86
6.2	Algoritmus lokálního prohledávání	93
6.3	Evoluční algoritmus	101
6.3.1	Volba reprezentace	102
6.3.2	Oprava chromosomu	103
6.3.3	Selekce	105
6.3.4	Křížení	106
6.3.5	Mutace	107
6.3.6	Volba nastavení operátorů velikost populace, selekce a křížení	109
6.3.7	Volba nastavení mutace hybridizací lokálním prohledáváním	119
6.3.8	Testování zvoleného nastavení na modelech JSSP a FJSSP	122
6.4	Automatická optimalizace	127
6.5	Dílčí závěr č.4 – Zhodnocení výsledků heuristických algoritmů aplikovaných na etalonových modelech	134
7	Ověření metodiky využití rozvrhovacích metod na modelech reálné výroby	136
7.1	TOS VARNSDORF a.s.	136
7.1.1	Definování požadavku na plán – rozvrh	136
7.1.2	Stanovení měřitelných ukazatelů	137
7.1.3	Analýza současného stavu	138

7.1.4	Testování modelu pomocí navržených rozvrhovacích metod	144
7.1.5	Navržení opatření vzhledem k průběhu testování navržené metodiky....	148
7.2	MZ Liberec a.s.	149
7.2.1	Definování požadavku na plán – rozvrh.....	150
7.2.2	Stanovení měřitelných ukazatelů.....	150
7.2.3	Analýza současného stavu	151
7.2.4	Testování modelu pomocí navržených rozvrhovacích metod	154
7.2.5	Navržení opatření vzhledem k výsledkům testů.....	158
7.3	Dílčí závěr č.5 – Zhodnocení ověření metodiky využívání heuristických algoritmů v rozvrhování výroby	160
8	Závěrečné shrnutí disertační práce	162
8.1	Shrnutí cílů disertační práce.....	162
8.2	Teoretické přínosy disertační práce	165
8.3	Praktické přínosy disertační práce	165
8.4	Směry dalšího výzkumu.....	165

Seznam zkratek

Zkratka	Cizojazyčný význam	Český význam
A	Active schedule	Aktivní rozvrhy
ACO	Ant Colony Optimization	Optimalizace mravenčích kolonií
AIS	Artificial Immune Systems	Umělé imunitní systémy
AO		Automatická optimalizace
AP	Assignment Problem	Přiřazovací problém
APS	Advanced Planning and Scheduling	Pokročilé plánování a rozvrhování
APSS	Adaptive Population Size Schemes	Adaptivní algoritmus manipulující s velikostí populace
BB	Bucket Brigades	Hašení ohně (kbelíkové brigády)
BPP	Bing Packing Problem	Problém balení batohů
CA	Constructive Algorithm	Konstruktivní algoritmus
CNC	Computer Numeric Control	Číslicově řízený počítačem
CP	Critical Path	Kritická cesta
CPA	Critical Path Analysis	Analýza kritické cesty
CPM	Critical Path Method	Metoda kritické cesty
CPU	Central Processing Unit	Centrální výpočetní jednotka
DBR	Drum - Buffer - Rope	Systém řízení výroby pomocí TOC
DMAIC	Define, Measure, Analyse, Improve, Control	Definuj, měř, analyzuj, zlepšuj, kontroluj
DŘV		Dílenské řízení výroby
E	Elitism	Elitářská selekce
EA	Evolution Algorithm	Evoluční algoritmus
ECC	Elitism Clone Control	Elitářská selekce s kontrolou klonů
ERP	Enterprise Resource Planning	Celopodnikové plánování zdrojů
ES	Evolution Strategy	Evoluční strategie
FCFS	First Come First Served	První příchozí je prvním obsluženým
FFSSP	Flexible Flow Shop Scheduling Problem	Flexibilní proudový rozvrhovací problém
FIFO	First In, First Out	První vstupující je prvním vystupujícím
FJSSP	Flexible Job Shop Scheduling Problem	Flexibilní sekvenční rozvrhovací problém
FOSSP	Flexible Open Shop Scheduling Problem	Flexibilní otevřený rozvrhovací problém
FSSP	Flow Shop Scheduling Problem	Proudový rozvrhovací problém
GA	Genetic Algorithm	Genetický algoritmus

Zkratka	Cizojazyčný význam	Český význam
GP	Genetic Programming	Genetické programování
HSM	Hyper Speed Machining	Vysoko rychlostní obrábění
JIT	Just In Time	Právě včas
JSSP	Job Shop Scheduling Problem	Sekvenční rozvrhovací problém
KP	Knapsack Problem	Problém balení batohu
LAP	Linear Assignment Problem	Lineární přiřazovací problém
LAPT	Longest Alternative Processing Time	Nejdelší alternativní procesní čas
LPT	Longest Processing Time	Nejdelší procesní čas
LS	Local Search	Lokální prohledávání
LWKR	Least WorK Remaining	Nejméně práce zbývá
MCSSP	Multi-component scheduling problem	Více komponentní rozvrhovací problém
MES	Manufacturing Execution System	Výrobní informační systém
MOST	Maynard Operation Sequence Technique	Maynardova technika analýzy pomocí sekvencí pohybů
MRP	Material Requirements Planning	Plánování materiálových požadavků
MRP II	Manufacturing Resource Planning	Plánování výrobních zdrojů
MRP III	Manufacturing Resource Planning	Plánování a rozvrhování podnikových zdrojů
MTM	Methods Time Measurement	Metoda analýzy pohybů
MWKR	Most WorK Remaining	Nejvíce práce zbývá
ND	Non Delay schedule	Rozvrhy bez zpoždění
NP		Nepolynomiální
OPFC	One Piece Flow Caravan	Tok jednoho kusu - karavana
OPT	Optimized Production Technology	Informační systém využívající TOC
OSSP	Open Shop Scheduling Problem	Otevřený rozvrhovací problém
P		Polynomiální
PDCA	Plane, Do, Check, Act	Plánuj, udělej, zkontroluj, jednej
PLM	Product Lifecycle Management	Správa životního cyklu výrobku
POLCA	Paired-cell Overlapping Loops of Cards with Authorization	Systém řízení výroby pomocí samoregulačních řídicích okruhů
PRoFIGA	Population Resizing on Fitness Improvement GA	Změna velikosti populace na základě zlepšení účelové funkce
PSO	Particle Swarm Optimization	Optimalizace rojení částic
PvCH	Parent versus Childe	Selekce rodičů proti dětem

Zkratka	Cizojazyčný význam	Český význam
RAM	Random Access Memory	Paměť s přímým vstupem
RS	Random Select	Náhodný výběr
RS_a	Random Select average	Náhodný výběr - průměrná hodnota
RS_b	Random Select best	Náhodný výběr nejlepší hodnota
RS_w	Random Select worst	Náhodný výběr - nejhorší hodnota
SD	Scheduling Diagrams	Rozvrhovací diagramy
SFHA	Sheep Flock Heredity Algorithm	Optimalizace ovčích stád
SMART	Specific, Measurable, Attainable, Relevant, Time-bound	Specifický, Měřitelný, Dosažitelný, Relevantní, Termínovaný
SME	Small and Medium Enterprises	Malé a střední podniky
SPT	Shorter Processing Time	Nejkratší procesní čas
SSP	Shop Scheduling Problem	Rozvrhovací problém
TOC	Theory of constraints	Teorie omezení
TPS	Toyota Production System	Výrobní systém Toyota
TPV		Technologická příprava výroby
TSP	Traveling Salesman Problem	Problém obchodního cestujícího
VBA	Visual Basic for Application	Jazyk pro tvorbu aplikací
VRP	Vehicle Routing Problem	Problém okružních jízd
WIP	Work In Process	Rozpracovaná výroba
XSSP	Mixed Shop Scheduling Problem	Smíšený rozvrhovací problém

Seznam symbolů

Symbol	Výklad symbolu	Oblast
$\tau_{i,j}$	množství feromonu mezi uzly	optimalizace - mravenčí kolonie
$\eta_{i,j}$	požadavek na hranu	optimalizace - mravenčí kolonie
$\Delta \tau_{i,j}^k$	množství naneseného feromonu.	optimalizace - mravenčí kolonie
$\tau_{i,j}^*$	množství feromonu na dané hraně	optimalizace mravenčí kolonie
<i>Block</i>	blokování pracoviště	rozvrhování – omezení modelu
br_t	délka přestávky	rozvrhování operací
br_{te}	suma všech časů přestávek a nevýrobních směn, které se vyskytnou v následující iteraci rozvržení operace (zakázky)	rozvrhování operací
c_i	učicí faktor	optimalizace - rojení částic
C_{max}	maximální čas dokončení všech zakázek	hodnocení účelové funkce
<i>Contr</i>	cena za zkrácení trvání operace	rozvrhování – omezení modelu
d_i	termín dokončení zakázky	rozvrhování operací
d_i	učicí faktor	optimalizace - rojení částic
e_{ij}	termín zahájení operace	rozvrhování operací
e_t	nejdříve možný čas, kdy operace o_k v S_t může začít.	rozvrhování operací
e_{tr}	absolutní počáteční začátek operace - zakázky ve vybraných časových jednotkách (e.g. 30 [min])	rozvrhování operací
f	účelová funkce	hodnocení účelové funkce
F	vhodnost řešení	optimalizace - genetické algoritmy
FR	faktor růstu	změna velikosti populace
f_t	nejdříve možný čas, kdy operace o_k v S_t může skončit	rozvrhování operací
f_{tr}	relativní čas konce zakázky (operace)	rozvrhování operací
<i>CH</i>	počet operací	rozvrhování – omezení modelu
inimaxFIT	nejlepší hodnota účelové funkce u první generace.	změna velikosti populace
J_i	označení zakázky	modely rozvrhovacích problémů
L_{max}	maximální zpoždění všech zakázek	rozvrhování – omezení modelu
m	počet strojů	modely rozvrhovacích problémů
maxFITn	nejlepší současná hodnota účelové funkce.	změna velikosti populace
maxFITp	předchozí nejlepší hodnota účelové funkce.	změna velikosti populace
MG	maximální počet generací	změna velikosti populace
M_j	označení stroje	modely rozvrhovacích problémů
m_t	absolutní čas nejdříve možného začátku operace (zakázky) o_k	rozvrhování operací
m_{tr}	relativní čas nejdříve možného začátku operace (zakázky)by o_k	rozvrhování operací
N	počet zakázek	modely rozvrhovacích problémů

Symbol	Výklad symbolu	Oblast
N	velikost populace	optimalizace – evoluční algoritmy
$N(x)$	množina (sousedních)řešení	složítost rozvrhovacích problémů
n	počet operací	modely rozvrhovacích problémů
$nowait$	bezprostřední začátek operace	rozvrhování – omezení modelu
O	časová složitost	složítost rozvrhovacích problémů
P	Metropolisovo kritérium přijetí řešení	optimalizace – simulované žíhání
$P_{i,j}$	pravděpodobnost, se kterou se mravenec rozhoduje pro přijetí trasy	optimalizace – mravenčí kolonie
P_t	částečný rozvrh ($t-1$) rozvržených operací	rozvrhování operací
p	stávající částice	optimalizace – rojení částic
p_i	počet dominantních řešení	Pareto optimalita
p_{ij}	procesní čas	rozvrhování operací
$prmp$	závislost operace na operaci předchozí – možnost přerušení operace	rozvrhování – omezení modelu
$prmtn$	permutační omezení trajektorie výrobním systémem	rozvrhování – omezení modelu
q	nové částice	optimalizace – rojení částic
r	počet operací s nenulovým procesním časem	rozvrhování – omezení modelu
$Recrc$	opakované zpracování na téže pracovišti	rozvrhování – omezení modelu
r_i	omezení začátku operace	rozvrhování – omezení modelu
R_t	množina všech operací, které jsou první v sekvenci generované náhodným klíčem	rozvrhování operací
$S(x)$	množina změn	složítost rozvrhovacích problémů
se_k	délka seřízení délka seřízení o_k	rozvrhování operací
sf_t	délka směny	rozvrhování operací
$sft_m\{ \}$	dostupná směna na stroji	rozvrhování operací
sft_t	relativní číslo směny (například 1. směna –dle e_{tr} a m_{tr})	rozvrhování operací
SG	pořadové číslo současné generace.	změna velikosti populace
S_{sd}	závislost na seřizovacím čase	rozvrhování operací
S_t	množina operací, které lze rozvrhovat v kroku t , tzn. všechny operace, které předchází operacím v S_t , jsou v P_t	rozvrhování operací
T	teplota – řídicí parametr	optimalizace – metoda simulovaného žíhání
$tree$	závislost na operaci jiné zakázky	rozvrhování – omezení modelu
w	váha hodnocení kritéria	hodnocení účelové funkce
x	reálné rozvrhy	složítost rozvrhovacích problémů
α	parametr kontroly vlivu na $\tau_{i,j}$	optimalizace – mravenčí kolonie

Symbol	Výklad symbolu	Oblast
α	hladina významnosti	statistika
α	základní trajektorie zakázky přes stroje	notace rozvrhovacích modelů
B	parametr kontroly vlivu na $\eta_{i,j}$ ($\beta \leq 1$)	optimalizace – mravenčí kolonie
β	určuje charakteristika zpracování	notace rozvrhovacích modelů
γ	specifikace optimalizačního kritéria	notace rozvrhovacích modelů
δ	definovaná vzdálenost	Optimalizace – rojení částic
λ	počet jedinců – potomků	evoluční strategie
μ	počet jedinců ve stávající populaci	evoluční strategie
ρ	poměr vypařování feromonu	optimalizace – mravenčí kolonie
ρ	počet rodičů potomka	evoluční strategie
ΣC_i	celkový čas všech zakázek	hodnocení účelové funkce
ΣE_i	celkový čas brzkého dodání	hodnocení účelové funkce
ΣT_i	celkové opoždění všech zakázek	hodnocení účelové funkce
ΣU	celkové penále za nedodržení požadavku s ohledem na zakázku	hodnocení účelové funkce

Seznam obrázků

Obr. 1-1: Optimalizace pořadí výroby dílců	21
Obr. 2-1: Informační tok výrobním systémem [29].....	28
Obr. 3-1: Eulerův diagram pro P, NP, NP těžké a úplné problémy	30
Obr. 3-2: Diagram obecného otevřeného problému (OSSP)	33
Obr. 3-3: Flexibilní model otevřeného rozvrhovacího problému (FO2)	34
Obr. 3-5: Model flexibilního proudového problému (FFSSP)	35
Obr. 3-6: Model sekvenčního rozvrhovacího problému (JSSP)	37
Obr. 3-7: Model flexibilního sekvenčního problému (FJSSP)	38
Obr. 3-8: Smíšený rozvrhovací problém (XSSP)	39
Obr. 3-9: Model více komponentního rozvrhovacího modelu (MCSSP).....	40
Obr. 4-1: Metody pro tvorbu a optimalizaci dílenských rozvrhů	44
Obr. 4-2: „Neaktivní“ rozvrh	48
Obr. 4-3: Semi-aktivní rozvrh.....	48
Obr. 4-4: Aktivní rozvrh	49
Obr. 4-5: Rozvrh bez zpoždění	49
Obr. 4-6: Mapa rozvrhů	49
Obr. 4-7: Částečný strom rozvrhovacího problému.....	51
Obr. 4-8: Příklad horolezeckého algoritmu na jednoduché funkci [107]	56
Obr. 4-9: Postup při optimalizaci GA	60
Obr. 4-10: Jednobodové křížení.....	61
Obr. 5-1: Postup při definici požadavků na plán rozvrh.....	76
Obr. 5-2: Postup stanovení měřitelných ukazatelů	78
Obr. 5-3: Analýza současného stavu.....	81
Obr. 5-4: Optimalita a výpočetní čas heuristických metod	82
Obr. 5-5: Testování modelu výrobního systému a aplikace metod rozvrhování	83
Obr. 5-6: Kroky při zvážení přínosů zavedení rozvrhovacích metod	84
Obr. 5-7: Hlavní vazby kroků navržené metodiky	85
Obr. 6-1: Procedura prohledávání okolí jedním prohozením	93
Obr. 6-2: Diagram algoritmu založeného na lokálním prohledávání	94
Obr. 6-3: Diagram evolučního algoritmu	101
Obr. 6-4: Reprezentace pomocí náhodného klíče	102

Obr. 6-5: Reprezentace pomocí náhodného klíče pro JSSP a FJSSP	103
Obr. 6-6 : Oprava chromosomu	104
Obr. 6-7: Ukázka uniformního křížení	107
Obr. 6-8: Dopředný rozvrh	108
Obr. 6-9: Zpětný rozvrh po fixaci na dosažený makespan	108
Obr. 6-10: Změna sekvence operací na kritickém bloku	109
Obr. 6-11: Postup tvorby rozvrhu pomocí automatické optimalizace	130
Obr. 7-1: Příklad trajektorie výrobním systémem TOS VARSDORF a.s.	139
Obr. 7-2: Ganttův graf seřizování bez potřeby kusů z předchozí operace.....	141
Obr. 7-3: Postup úpravy startovních a koncových časů výrobních dávek dle směnnosti	143
Obr. 7-4: Ganttův diagram - ukázka pohledu přes stroje.....	145
Obr. 7-5: Ganttův diagram - ukázka pohledu přes zakázky	145
Obr. 7-6: Příklad trajektorie výrobním systémem MZ Liberec a.s.	153
Obr. 7-7: Ganttův diagram - ukázka pohledu přes stroje MZ Liberec a.s.	155
Obr. 7-8: Ganttův diagram - ukázka pohledu přes zakázky	155

Seznam tabulek

Tab. 3-1: Přehled aplikací modelů OSSP, FSSP, JSSP a jejich flexibilních modifikací.	42
Tab. 4-1: Model sekvenčního problému	48
Tab. 4-2: Chromozom jednoho jedince	60
Tab. 4-3: Ohodnocení populace	60
Tab. 4-4: Pravděpodobnost výběru jedince pro reprodukci ruletovým pravidlem.	61
Tab. 4-5: Genotyp genetického algoritmu	62
Tab. 4-6: Populace v rámci genetického programování	63
Tab. 4-7: Porovnání GA a AIS v analogii k rozvrhování [136]	67
Tab. 4-8: Přesné metody	70
Tab. 4-9: Přibližné metody – konstruktivní algoritmy	70
Tab. 4-10: Přibližné metody – metody založené na lokálním prohledávání	71
Tab. 4-11: Evoluční algoritmy	71
Tab. 4-12: Metody umělé inteligence	72
Tab. 6-1: Množina všech řešení testovaných modelů JSSP.....	88
Tab. 6-2: Odhad velikosti množiny všech řešení testovaných modelů FJSSP	89
Tab. 6-3: Výsledky prioritních pravidel – makespan (aktivní rozvrh modely JSSP).....	89
Tab. 6-4: Výsledky prioritních pravidel – makespan (rozvrh bez zpoždění modely JSSP)	89
Tab. 6-5: Výsledky prioritních pravidel – makespan (aktivní rozvrh modely FJSSP)...	90
Tab. 6-6: Výsledky prioritních pravidel – makespan (rozvrh bez zpoždění modely FJSSP).....	90
Tab. 6-7: Odchylka od optima prioritních pravidel – makespan (aktivní rozvrh modely JSSP).....	90
Tab. 6-8: Odchylka od optima prioritních pravidel – makespan (rozvrh bez zpoždění modely JSSP).....	91
Tab. 6-9: Odchylka od optima prioritních pravidel – makespan (aktivní rozvrh modely FJSSP).....	91
Tab. 6-10: Odchylka od optima prioritních pravidel – makespan (rozvrh bez zpoždění)	92
Tab. 6-11: Časová náročnost rozvrhování JSSP	92
Tab. 6-12: Časová náročnost rozvrhování	93
Tab. 6-13: Hodnoty účelové funkce makespan použitím LS – aktivní rozvrhy JSSP	94

Tab. 6-14: Hodnoty účelové funkce makespan použitím LS –rozvrhy bez zpoždění JSSP	95
Tab. 6-15: Hodnoty účelové funkce makespan použitím LS – aktivní rozvrhy FJSSP..	95
Tab. 6-16: Hodnoty účelové funkce makespan použitím LS –rozvrhy bez zpoždění FJSSP	95
Tab. 6-17: Zlepšení účelové funkce makespan použitím LS – aktivní rozvrhy JSSP	96
Tab. 6-18: Zlepšení účelové funkce makespan použitím LS –rozvrhy bez zpoždění JSSP	96
Tab. 6-19: Zlepšení účelové funkce makespan použitím LS – aktivní rozvrhy FJSSP..	97
Tab. 6-20: Zlepšení účelové funkce makespan použitím LS –rozvrhy bez zpoždění FJSSP	97
Tab. 6-21: Vzdálenost nalezeného makespan od optima – aktivní rozvrhy JSSP	98
Tab. 6-22: Vzdálenost nalezeného makespan od optima –rozvrhy bez zpoždění JSSP .	98
Tab. 6-23: Vzdálenost nalezeného makespan od optima – aktivní rozvrhy FJSSP.....	98
Tab. 6-24: Vzdálenost nalezeného makespan od optima –rozvrhy bez zpoždění FJSSP	99
Tab. 6-25: Porovnání časového přínosu optimalizace LS - JSSP.....	99
Tab. 6-26: Porovnání časového přínosu optimalizace LS - FJSSP.....	100
Tab. 6-27: Nastavení parametrů EA v jednotlivých analýzách	110
Tab. 6-28: Výsledky testování FT10 jednoduchý elitismus	112
Tab. 6-29: Výsledky testování MK10 jednoduchý elitismus	113
Tab. 6-30: Výsledky testování FT10 ECC.....	113
Tab. 6-31: Výsledky testování MK10 ECC	114
Tab. 6-32: Výsledky testování FT10 PvCH.....	114
Tab. 6-33: Výsledky testování MK10 PvCH.....	114
Tab. 6-34: Výsledky testování FT10 - hybridizace	119
Tab. 6-35: Výsledky testování MK10 - hybridizace.....	119
Tab. 6-36: Výsledky aktivních rozvrhů účelové funkce makespan JSSP.....	122
Tab. 6-37: Výsledky rozvrhů bez zpoždění účelové funkce makespan JSSP	123
Tab. 6-38: Výsledky aktivních rozvrhů účelové funkce makespan FJSSP	123
Tab. 6-39: Výsledky rozvrhů bez zpoždění účelové funkce makespan FJSSP	124
Tab. 6-40: Zlepšení aktivních rozvrhů účelové funkce makespan JSSP pomocí EA...	124
Tab. 6-41: Zlepšení rozvrhů bez zpoždění účelové funkce makespan JSSP pomocí EA.	125
Tab. 6-42: Zlepšení aktivních rozvrhů účelové funkce makespan FJSSP pomocí EA.	125

Tab. 6-43: Zlepšení rozvrhů bez zpoždění účelové funkce makespan FJSSP pomocí EA.	125
Tab. 6-44: Porovnání časového přínosu optimalizace EA - JSSP	126
Tab. 6-45: Porovnání časového přínosu optimalizace EA - FJSSP	126
Tab. 6-46: Porovnání EA a AO modelů JSSP – aktivní rozvrhy	131
Tab. 6-47: Porovnání EA a AO modelů JSSP – rozvrhy bez zpoždění	132
Tab. 6-48: Porovnání EA a AO modelů FJSSP – aktivní rozvrhy	132
Tab. 6-49: Porovnání EA a AO modelů FJSSP – rozvrhy bez zpoždění	132
Tab. 7-1: Ilustrace transportní matice	144
Tab. 7-2: Náhled kontroly 1/3 modelu 1. komponentu hlavy HUR	144
Tab. 7-3: Hodnoty účelové funkce makespan generované pomocí CA 1/3 modelu TOS VARNSDORF a.s.	146
Tab. 7-4: Hodnoty makespan dle automatické optimalizace 2/3 modelu TOS VARNSDORF a.s.	146
Tab. 7-5: Hodnoty makespan dle automatické optimalizace plného modelu TOS VARNSDORF a.s.	147
Tab. 7-6: Náhled kontroly 1/3 modelu MZ Liberec a.s.	154
Tab. 7-7: Hodnoty účelové funkce makespan generované pomocí CA 1/3 modelu MZ Liberec a.s.	156
Tab. 7-8: Hodnoty účelové funkce makespan generované pomocí AO 2/3 modelu MZ Liberec a.s.	156
Tab. 7-9: Hodnoty účelové funkce makespan generované pomocí AO plného modelu MZ Liberec a.s.	157
Tab. 7-10: Hodnoty účelové funkce makespan generované pomocí AO plného modelu MZ Liberec a.s.	157

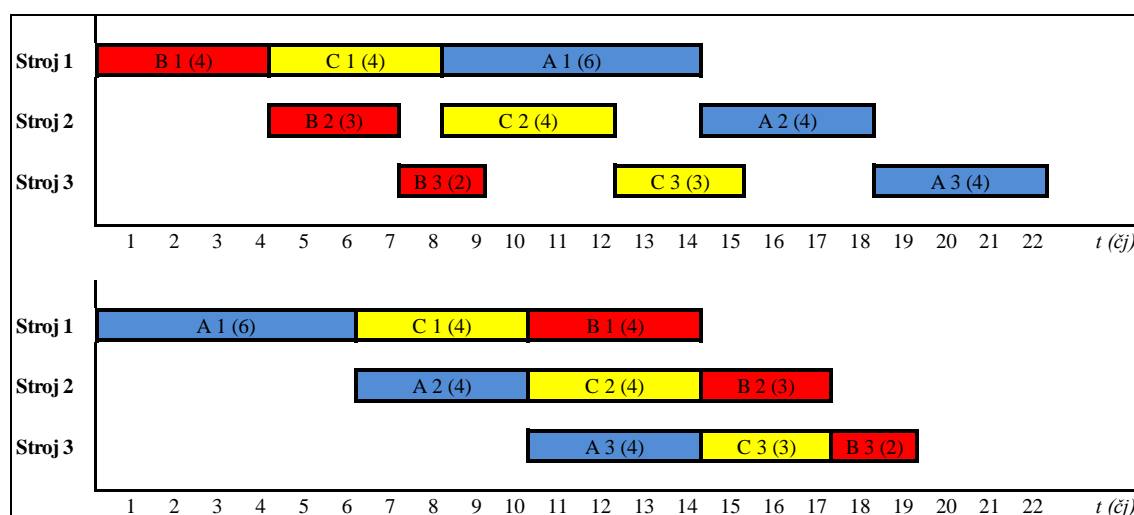
Seznam grafů

Graf 6-1: Model FT10 konvergence celé populace k jednomu řešení.....	111
Graf 6-2: Model FT10 – ukázka konvergence celé populace k jednomu řešení při 10% pravděpodobnosti křížení.....	111
Graf 6-3: Model FT10 ukázka populace u které nedošlo ke konvergenci všech jedinců při 40% pravděpodobnosti křížení.....	112
Graf 6-4: Model MK10 konvergence celé populace k jednomu řešení.....	113
Graf 6-5: Ověření normality dat účelové funkce makespan FT10	115
Graf 6-6: Ověření normality dat účelové funkce makespan MK10.....	115
Graf 6-7: Analýza polohy a variability – Krabicový graf funkce makespan modelu FT10	116
Graf 6-8: Analýza polohy a variability – Krabicový graf funkce makespan modelu MK10.....	116
Graf 6-9: Analýza vlivu křížení a selekce na makespan FT10	117
Graf 6-10: Analýza interakce vlivu křížení a selekce na makespan FT10	117
Graf 6-11: Analýza vlivu křížení a selekce na makespan MK10	118
Graf 6-12: Analýza interakce vlivu křížení a selekce na makespan MK10.....	118
Graf 6-13: Ověření normality dat účelové funkce makespan FT10	120
Graf 6-14: Ověření normality dat účelové funkce makespan FT10	120
Graf 6-15: Analýza polohy a variability – Krabicový graf makespan modelu FT10-hybridizace.....	121
Graf 6-16: Analýza polohy a variability – Krabicový graf makespan modelu MK10-hybridizace.....	121
Graf 6-17: 10 experimentů modelu FT10 pomocí AO	133

1 ÚVOD

Tvrdá konkurence a dynamické tržní prostředí nutí firmy k neustálému zlepšování zaváděním moderních technologií i organizací výroby. Řada firem svoje úsilí zaměřuje zejména na zeštíhlování svých procesů zaváděním metod štíhlé výroby. V dnešní době již však ani toto nestačí a je třeba hledat rezervy i v dalších oblastech podniku. Pozornost se tak obrací i na plánování a rozvrhování výroby. Tvorba plánu a následného detailního rozvrhu výrobních operací (resp. zakázek) v oblasti operativního a dílenského plánování výroby se stále podceňuje, přestože je zde značný potenciál pro zlepšování.

Důležitost diskutované problematiky dokumentuje následující příklad (obr. 1-1), kde je znázorněno, jak „pouhá“ změna pořadí výroby dílců může ovlivnit např. využití kapacit či průběžnou dobu výroby. Obrázek zobrazuje Ganttovy diagramy dvou rozvrhů s odlišným pořadím výroby dílců (A, B a C). Pořadí operace na dílci je uvedena za jménem dílce a časové náročnosti jsou uvedeny v závorce (např. druhá operace dílce C s časovou náročností 4 časové jednotky – C 2 (4)).



Obr. 1-1: Optimalizace pořadí výroby dílců

Problematika plánování a rozvrhování výroby však vůbec není triviální. Na výrobu jsou kladeny protichůdné požadavky, kde mimo dodržování termínů je snaha o maximální využití výrobních kapacit, co nejkratší průběžné doby a minimalizaci zásob. Je nutné řešit i konflikt omezených zdrojů, aktuální provozní stavy ve výrobě apod. Výběr metodiky řešení závisí i na velikosti podniku i dalších okolnostech.

Plánování a rozvrhování výroby se v dnešní době řeší převážně způsoby popsanými v následujících kapitolách.

1.1 Manuální rozvrhování

Manuální plánování a rozvrhování výroby je navzdory neustálému rozvoji softwarových řešení stále využíváno. Jedná se často o případy, kdy výrobní společnost nepovažuje plánování resp. rozvrhování za důležité, či se jedná o výrobní systémy natolik procesně jednoduché, že u nich není třeba využívat složitých výpočetních prostředků [1]. Takové výrobní systémy mají obvykle předmětné uspořádání výroby (linky či výrobní buňky – proudový výrobní systém) a jeden hlavní produkt, který nebývá vyráběn ve více variantách.

Takové plány resp. rozvrhy jsou pak založeny na:

- Průměrných průběžných časech výroby (např. dva týdny) neuvažujíc omezení výroby (úzká místa).
- Startovní časy jednotlivých výrobních zakázek jsou určeny dostupností materiálu a předpovědí dokončení předcházejících operací (ať již na stroji či na výrobní zakázce).

Hlavní činností v oblasti dílenského plánování a řízení je monitorování průběhu zakázek resp. jejich zpoždění vůči požadavku zákazníka a řízení výroby přirozeně pomocí prioritních pravidel dle zkušenosti s předchozími projekty. Nejčastěji je využíváno pravidlo „první vstupující je prvním vystupujícím“ (FIFO – First In First Out), které je zajištěno výrobními frontami před každým z pracovišť. Manuální rozvrhování pak ve většině případů vede k vysoké rozpracovanosti (WIP - Work In Process), zpoždění zakázek vzhledem k požadovaným termínům zákazníka a neustálému „hašení požárů“. To je pak prováděno udělováním vysokých priorit těm zakázkám, které mají vzhledem k datu doručení největší zpoždění. Toto přerozdělování priorit pak způsobuje lavinový efekt, kdy nejsou zpožděny jen původně opožděné zakázky, ale i zakázky, které byly původně bez zpoždění.

Obvyklým řešením je pak navyšování výrobních kapacit. To může vést právě jak u složitějších „zakázkových“ výrobních systémů (vysoké spektrum výrobků s odlišnou trajektorií výrobním systémem a s malým počtem kusů), tak i u jednodušších proudových, k složitějším sekvenčním problémům vzhledem k zvýšení počtu výrobních front a nutnosti určovat pořadí zakázek v nich.

V případě, že se jedná o „zakázkový“ výrobní systém, je možným řešením omezení spektra výrobků a transformace výrobního systému na linkovou výrobu. To na jedné straně omezí rozvrhovací problém – určíme pouze pořadí zakázek na vstupu do linky, na straně druhé se však sníží odolnost výrobní společnosti vůči výkyvům trhu.

Další možností pro malé a střední podniky je zavedení rozvrhovacích tabulí či softwarového nástroje, což však v řadě případů bývá nákladné jak časově (zřídka kdy je výrobní proces popsán dostatečně), tak finančně.

1.2 Rozvrhování pomocí plánovacích tabulí nebo tabulkových kalkulátorů

Velmi častým nástrojem malých a středních firem v oblasti plánování a rozvrhování, zvláště na úrovni dílenského řízení – rozvrhování, jsou plánovací tabule či tabulkové kalkulátory [2].

Tyto nástroje mohou být dostatečné pro malé a jednoduché výrobní systémy. Nedostačující pak bývají v případech, kdy je rozvrhováno velké množství výroby „zakázkového“ typu a to z důvodu složitého přerozvrhování.

Časté bývají aplikace elektronické rozvrhovací tabule v prostředí tabulkových kalkulátorů na bázi MS Excel. Tyto aplikace jsou navrženy k tvorbě reálných rozvrhů díky možnostem, které nabízí VBA (Visual Basic for Applications). Vzhledem k vysoké složitosti některých rozvrhovacích problémů „zakázkového“ typu, nejsou tyto aplikace schopné nalézt reálná řešení.

Tento problém si uvědomila řada softwarových společností a rozšířila koncept rozvrhovacích (plánovacích) tabulí na elektronickou verzi, která nabízí tvorbu a

modifikaci rozvrhu pomocí Ganttova grafu (např. dopředné a zpětné přeplánování resp. rozvrhování).

Hlavní nevýhodou těchto aplikací v případě rozvrhování většího množství zakázek s rozdílnou trajektorií výrobním systémem je vysoký čas pro generování rozvrhů splňujících kritéria společnosti a zákazníka, což je patrné zejména při použití optimalizačních technik (např. změna pořadí zakázek v softwarovém nástroji pomocí „drag&drop“).

1.3 Rozvrhování v oblasti materiálového plánování

V oblasti plánování a rozvrhování je již delší dobu populární implementace systémů pro celopodnikové plánování zdrojů (ERP – Enterprise Resource Planning), které pokrývají širokou škálu podnikových procesů díky modulům jako je účetnictví, skladové hospodářství, plánování nákupu materiálu, transport atd. Využitím modulů jako je „Dílenské řízení výroby (DŘV)“ se zlepšují parametry výrobního procesu (využití strojů, monitorování zásob, zkracování průběžné doby výroby atd.) pomocí integrace systému pro odvádění resp. monitorování výroby v centrální databázi, které přibližují reálný stav výroby řízenému modelu výroby. Systémy ERP, které obsahují DŘV, jsou tak schopny v závislosti na frekvenci aktualizace dat informovat interního zákazníka o aktuálním stavu zakázek. Plánování resp. rozvrhování je pak prováděno v závislosti na aplikovaném principu řízení výroby a to tlakem, tahem, či pomocí úzkých míst.

1.3.1 Řízení tlakem - ERP/APS a MES systémy

Vzhledem k tomu, že ERP jsou v oblasti plánování a rozvrhování výroby založeny na konceptu lhůtového plánování [3] (MRP – Material Requirements Planning) či častěji kapacitního plánování [4] (MRP II - Manufacturing Resource Planning), nejsou tyto schopné rozvrhovat výrobu do omezených kapacit. Výstupem takového systému pak bývá fronta resp. seznam zakázek, které mají být v daném časovém horizontu zpracovány a to bez údaje, v jaké sekvenci. Tento problém je obvykle řešen ručním rozvrhováním, rozvrhováním pomocí tabulkových kalkulátorů [5], systémy MRP III [6], a v poslední době pomocí systémů pro pokročilé plánování a rozvrhování čili APS (Advanced Planning and Scheduling).

I u těchto systémů může docházet ke generování nereálných rozvrhů vzhledem k tomu, že data získávaná z technologické přípravy výroby (TPV) nemusí být dostatečná (např. chybí údaje o transportu, druhu seřízení atd.). Dílenští mistři pak musí v prostředí elektronických plánovacích tabulí tyto rozvrhy opravovat způsobem přetahování výrobních zakázek do jiných časových horizontů či výrobních zdrojů [7]. Bez znalosti principů rozvrhování je tato činnost značně zdlouhavá.

Řešením může být přizpůsobení obecného modelu výrobního systému ERP dalším omezením v nástroji APS, to však bývá často jak časově, tak finančně náročné.

Složitější ERP/APS, jsou schopny rozvrhovat výrobní zakázky pomocí prioritních pravidel, jako jsou např. nejkratší procesní čas, nejvíce práce zbývá či časový zákaznický požadavek, či pomocí evolučních algoritmů [8]. Na jedné straně tyto systémy poskytují reálné plány díky uvažování definovaných omezení a nemožnosti zasáhnout do procesu rozvrhování, na straně druhé tím znemožňují následnou optimalizaci v dílenském prostředí.

Velkým problémem APS je pak oblast použití. APS jsou používány pro dlouhodobé plánování vzhledem k ceně takového systému a délce trvání tvorby rozvrhu pro celou výrobní oblast. S ohledem na to, že dílenští plánovači mohou jen částečně ovlivnit rozvrh celé výroby, vznikl požadavek na systém, jenž by byl vhodný pro dílenské řízení výroby a nebyl pouze systémem pro monitorování materiálového toku (příjem a výdej materiálu, odvádění hotové výroby, evidence zmetkovitosti atd.).

Takovýto požadavek se v současné době snaží plnit výrobní informační systémy MES (Manufacturing Execution Systems), které vycházejí z datové základny ERP stejně jako tomu je u APS [9]. Ty splňují požadavky v oblasti monitorování materiálového toku a aktuálního stavu výrobních zařízení i optimalizace dílenského rozvrhu pomocí prostředí elektronických plánovacích tabulí.

1.3.2 Řízení tahem - štíhlá výroba

Produkční systém Toyota (TPS - Toyota Production System) je světově známým systémem, který zavedl do praxe mnoho konceptů, principů a metod. Tento systém se spíše než na plánování a rozvrhování zaměřuje na samotné řízení výrobního procesu [10]. Heijunka (vyrovnání výroby prostřednictvím objemu a skladby sortimentu) spolu s Kanbanem (samořídící regulační okruhy) je vhodným nástrojem pro řízení opakované výroby. Pomocí regulace materiálového toku v celém podniku se pak předchází vyhladovění úzkých míst, zmenšují se zásoby, dodržují zákaznické termíny atd.

Systémy založené na Kanbanu však nejsou vhodné na výroby „zakázkového“ typu. Kanban neposkytuje informaci o následném materiálovém toku, předpokladu ukončení operace ve výrobní zakázce v případě, kdy zakázky mají rozdílnou trajektorii výrobním systémem. Tento problém byl řešen např. pomocí dvou-kartového systému [11] či systému POLCA [12, 13]. Tímto způsobem byla vyřešena problematika trasování zakázek, avšak ne optimalizace výrobního plánu např. pomocí optimalizace sekvence výrobních zakázek či volba alternativních výrobních zdrojů.

1.3.3 Řízení pomocí úzkých míst - teorie omezení

Jedním z nástrojů pro plánování a rozvrhování výroby je nástroj Buben – Zásobník – Lano (DBR – Drum – Buffer – Rope), jenž vychází z teorie omezení (TOC - Theory Of Constraints). Tato teorie vychází z předpokladu, že každý typ výroby je omezen jedním zdrojem omezení a ostatní zdroje mají dostatečnou kapacitu na to, aby uspokojily jakýkoli rozvrh na zdroji omezení. To znamená, že nástroj DBR se v oblasti rozvrhování zabývá pouze tzv. úzkým místem ignorujíc kapacity všech ostatních tzv. neúzkých míst. V řadě případů se však při použití této teorie vyskytují takzvaná putující úzká místa.

Jedním z optimalizačních nástrojů oblasti materiálového plánování je tzv. systém Optimalizované Produkční Technologie [14] (OPT - Optimized Production Technology), který využívá principů nástroje DBR. OPT systém optimalizuje výrobní plán pomocí volby velikosti výrobních dávek, sekvence výrobních zakázek na úzkém místě vzhledem k proměnné délce seřizovacích časů a volbou alternativních výrobních zdrojů.

Problém putujících úzkých míst je pak řešen některým z algoritmů známých z operačního výzkumu zaměřujících se na tuto problematiku. [15, 16, 17, 18]. Přesto v případech „zakázkového“ typu výroby bývá složité z hlediska frekvence změny úzkého místa toto lokalizovat, což následně vede k redukci použití této metody na

rozvrhovací problém. Tzn. úzkým místem se stává více méně každé pracoviště a je tedy pro ně vytvářen detailní rozvrh.

1.4 Řízení s podporou počítačové simulace

Simulace je jedním z nejpoužívanějších nástrojů jak v oblasti návrhu výrobních systémů, tak s dlouhou historií v oblasti rozvrhování výroby [19,20]. Simulace je také používána v oblasti co-když analýz díky schopnosti popsat daný výrobní systém do úrovně možností a schopností uživatele. Její hlavní nevýhodou je pak složitá návaznost na materiálové plánování vzhledem k vazbám na TPV resp. ERP systémy.

V současné době je simulace implementována do systémů celoživotního cyklu výrobku (PLM - Product Lifecycle Management) jako je Delmia či Tecnomatic, spolu s možností optimalizovat výrobu v oblasti kombinatorických problémů jako je volba pořadí výrobních zakázek. Přestože je v simulaci možnost rozvrhování stále vzácná, jsou známy možné postupy jak simulaci v této oblasti využít nejen v případě výrobních systémů typu linka či výrobní buňka [21]. Hlavní výhodou pak zůstává možnost modelace a optimalizace velkého spektra výrobních systémů „zakázkového“ typu (hybridní, flexibilní) [22, 23, 24]. Nevýhodou simulace v oblasti rozvrhování však zůstává to, že je schopna vlastními prostředky generovat pouze rozvrhy typu „bez zpoždění“ o kterých je známo, že nemohou díky své podstatě vždy nalézt optimální řešení.

1.5 Shrnutí

Jak bylo popsáno, plánovací a rozvrhovací problémy je možné řešit mnoha způsoby, které mají své výhody a nevýhody:

- Ruční rozvrhování
 - Výhody – levné, flexibilní.
 - Nevýhody – při větším množství zakázek v podstatě nemožné z časových důvodů najít „optimální“ plán dle daného kritéria a dochází k problému „hašení ohně“.
- Rozvrhování pomocí plánovacích tabulí nebo tabulkových kalkulátorů
 - Výhody – levné, prostředí tabulkových kalkulátorů je pro dílenské plánovače známé, rychlejší a přehlednější než v případě ručního rozvrhování. Možnost použití rozvrhovacích algoritmů.
 - Nevýhody – omezená možnost použití algoritmů vzhledem k podstatě VBA.
- Řízení tlakem - ERP/APS a MES systémy
 - Výhody – přímá vazba na materiálové plánování a nevýrobní procesy ve společnosti.
 - Nevýhody – zpravidla pouze plánování do neomezených kapacit (v případě ERP), malá možnost ovlivňovat rozvrh dílenským plánovačem. Časově a ekonomicky náročné.

- Řízení tahem
 - Výhody – samořídící systém, který potřebuje minimální úsilí plánovat.
 - Nevýhody – prakticky neaplikovatelné v prostředí výroby „zakázkového“ typu.
- Teorie omezení
 - Výhody – redukce problému na řízení úzkého místa, maximální využití „selského“ rozumu oproti ostatním metodám, orientace na výkon
 - Nevýhody – Problém putujících úzkých míst, obtížně aplikovatelné na prostředí výroby zakázkového typu.
- Simulace
 - Výhody – možnost brát v úvahu všechna omezení ve výrobě, co-když analýzy,
 - Nevýhody – nutná znalost simulačního softwaru, cena, nemožnost produkovat aktivní plány.

Velkým problémem jednotlivých způsobů plánování a rozvrhování výroby však stále zůstává zejména to, že je kladen velmi malý důraz na optimalizaci pořadí zakázek. Přes razantní vývoj v této oblasti, který potvrzuje značný potenciál pro řešení rozvrhovacích problémů pomocí optimalizace pořadí výrobních zakázek (viz např. [25, 26, 27, 28]), je to patrné zejména v průmyslové praxi.

To může být způsobeno následujícími důvody [7]:

- Modely výroby použité pro optimalizaci jsou ve většině případů zjednodušeny.
- Vývoj se zaměřuje na složité algoritmy, s účelem dosáhnout vždy co nejlepších optim.
- Důraz je kladen na nalezení optima bez důrazu na časovou náročnost optimalizace.
- Složitost algoritmů a s tím spojené nastavení optimalizačních parametrů je obtížně pochopitelné pro běžného dílenského plánovače.

2 CÍLE A STRUKTURA DISERTAČNÍ PRÁCE

Vzhledem k výše popsané problematice rozvrhování výrobních operací je cílem této disertační práce **navrhnout metodiku využívání heuristických algoritmů v rozvrhování výroby v SME společně s optimalizačním nástrojem rozvrhování výrobních operací pro potřeby dílenského resp. operativního rozvrhování.** Optimalizační nástroj bude splňovat následující:

- Využití rozvrhovacích optimalizačních metod.
- Nízká časová náročnost přeplánování.
- Možnost rozvrhování bez hluboké znalosti rozvrhovacích algoritmů.
- Zahrnutí běžných omezení ve strojírenské výrobě.
- Využití stávajících prostředků dostupných malému a střednímu podniku.

Hlavního cíle bude dosaženo těmito následujícími kroky:

1) Analýza problémů rozvrhování výroby, popis používaných výrobních modelů. (kap. 3).

Cílem je nalézt takový model, který by bylo možné použít pro co nejširší oblast rozvrhovacích problémů ve strojírenské výrobě.

2) Studium možností generování výrobních rozvrhů pro daný model včetně metod pro jejich optimalizaci. (kap. 4).

Cílem je nalezení metod využitelných pro návrh softwarové aplikace.

3) Navržení metodiky využití optimalizačních metod v oblasti rozvrhování výrobních zakázek. (kap. 5).

Cílem je navržení metodiky využití rozvrhovacího nástroje v oblasti strojírenské výroby.

4) Návrh, implementace a testování optimalizačních algoritmů ve vztahu ke známým optimálním řešením a časové náročnosti optimalizace. (kap. 6).

Cíle této části lze rozdělit do následujících tří skupin:

- Navržení algoritmu, který by vyhovoval podmínkám běžných omezení, které nejsou uvažovány v teoretických problémech, jako jsou transport, směnnost a seřizování stroje.
 - Ověření funkčnosti navrženého algoritmu z hlediska schopnosti nalézt co nejlepší řešení v požadovaném čase.
 - Navržení způsobu použití optimalizačních algoritmů bez nutnosti hluboké znalosti nastavení optimalizačních parametrů dílenským plánovačem.
- 5) Ověření navržené metodiky a optimalizačních algoritmů na reálných modelech strojírenské výroby pomocí softwarového nástroje (viz kap. 7).

3 OPTIMALIZAČNÍ PROBLÉMY A JEJICH SLOŽITOST

Cílem této kapitoly je popsat a analyzovat základní optimalizační algoritmy a jejich dělení s ohledem na řešenou problematiku kombinatorických problémů a vybrat takový, který je nejbližší problémům rozvrhování strojírenské výroby.

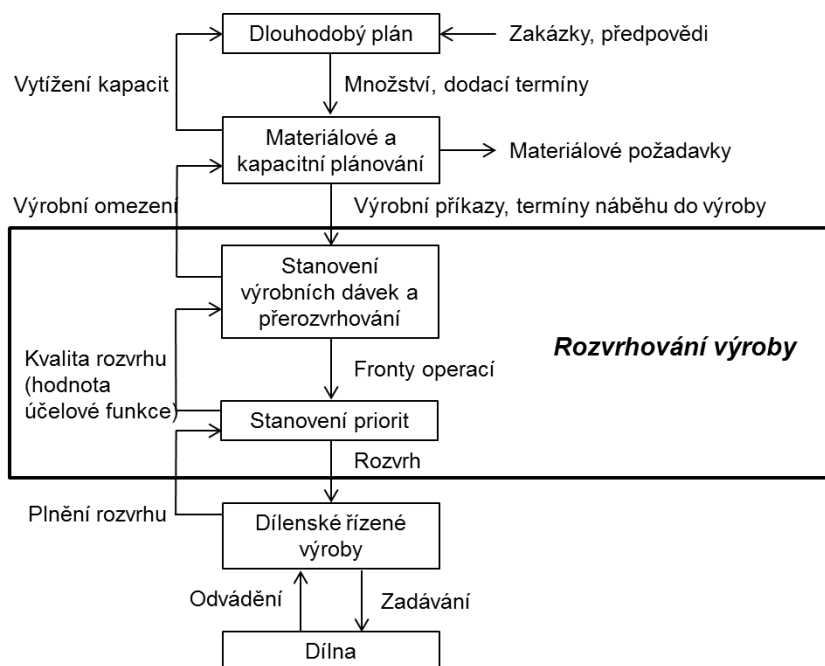
Proces rozvrhování výroby je ovlivněn mnoha vlivy, které jsou zpravidla závislé na způsobu rozvrhování, uplatňovaném ve firmě (manuální rozvrhování až použití počítačové simulace viz Úvod) a mohou se lišit případ od případu (např. princip řízení u tzv. „linkové výroby“ je jiný než u tzv. „zakázkové výroby“).

Ve výrobním systému musí zakázky projít plánovacím cyklem až do fáze rozpadu na jednotlivé operace se stanovenými termíny dokončení. Zakázky musí být obvykle zpracovány na strojích ve výrobních střediscích v sekvenci dané nejenom výrobním postupem, ale také rozvrhem výroby, který řeší základní konflikt požadavku zakázek na jeden stroj v jeden čas. V této oblasti dochází často k čekání výrobních zakázek na uvolnění zdroje, v extrémních případech i k situaci, kdy zakázka, která je již rozpracovaná na výrobním zdroji musí být přerušena z důvodu příchodu zakázky s vyšší prioritou.

Rozvrhování může být ovlivněno také plánovacím procesem, který pokrývá oblast tvorby dlouhodobého a střednědobého plánu. To se týká zejména vlivu zásob a předpovědí zákaznických poptávek, který ovlivňuje výrobní mix a tím alokaci zdrojů.

Rozvrhování je také ovlivněno nenadálými situacemi na dílně, jako jsou poruchy strojů, zvýšená zmetkovitost či nevhodně zvolené normo-časy operací. Tyto vlivy je možné v plánu, resp. rozvrhu zohlednit díky zpětné vazbě z výroby.

Výše zmíněné vlivy na plánování a řízení výroby stručně nastiňuje následující obr. 2-1 [29].



Obr. 2-1: Informační tok výrobním systémem [29]

Pro řešení optimalizačních problémů z oblasti plánování a řízení výroby existuje mnoho optimalizačních problémů [30] počínaje dopravní logistikou a výrobou samotnou konče.

Mezi nejznámější optimalizační problémy, v samotném rozvrhování výroby, patří například problém obchodního cestujícího (TSP) [31], který popisuje úlohu nalezení nejkratší cesty, s návratem na původní místo, mezi různými body na mapě s různou délkovou či časovou náročností na přemístění. V oblasti plánování a řízení výroby bývá tato úloha používána pro plánování trasy manipulačních prostředků.

Dalším známým problémem je tzv. problém batohu (KP) [32], kde je dána množina předmětů s danou hmotností a celková kapacita batohu. Cílem je nalézt takovou množinu předmětů, která nepřekročí kapacitu batohu a zároveň cena předmětů v batohu bude maximální. Rozšířením tohoto problému je problém naplňování zásobníku do kontejnerů (BPP) [33], kde je hlavním cílem uspořádat tvarově složité objekty do zásobníků tak, aby jich bylo potřeba co nejméně. Taková úloha je používána například při řešení nakládky žihací pece nebo tvorby nástřihového plánu.

Kombinací těchto (TSP a BPP) problémů se pak modeluje tzv. problém okružních jízd (VRP) [34], který popisuje obsluhu zákazníků na dané trase v rámci metody milk-run [35].

Mezi kombinatorické problémy používané přímo pro tvorbu dílenského rozvrhu patří zejména problémy přiřazovací (AP či LAP) a rozvrhovací (SSP).

AP modelují problematiku, kde jsou zdroje přiřazovány k úlohám s tím, že jakýkoli zdroj může být přiřazen k jakémukoli úkolu [36], avšak cena (nejčastěji čas) za splnění tohoto úkolu se bude lišit zdroj od zdroje. Klasickým případem uplatnění tohoto modelu je v oblasti přiřazování pracovníků v montážních či výrobních buňkách.

SSP pak řeší alokaci naplánovaných úloh na zdroje v čase. Jedná se o transformaci plánu, kdy je řešena otázka „co“ a „jak“, na rozvrh, který definuje „kde“ a „kdy“.

V oblasti plánování a rozvrhování výroby se lze setkat s optimalizačními problémy jak polynomiálními (P), tak nepolynomiálními (NP). Tyto problémy jsou rozebrány v následujících podkapitolách.

3.1 Časová složitost a třídy optimalizačních algoritmů

Optimalizační problémy v oblasti plánování a řízení výroby lze rozdělit podle časové složitosti¹ na dvě základní oblasti [37, 38, 39]. V oblasti klasifikace problémů se hovoří o dvou základních třídách problémů P a NP.

První třídou problémů jsou problémy řešitelné v polynomiálním čase (např. $O(n^k)$ kde k je konstanta) deterministickým polynomiálním algoritmem (P).

¹ Časová složitost $O(f(n))$ algoritmu řešícího problém o rozsahu n udává horní mez počtu $f(n)$ základních operací, které jsou pro vyřešení problému tímto algoritmem zapotřebí. V případě, kdy je časová složitost např. lineární, $O(n)$ se počet operací pro vyřešení tohoto problému zvýší tolikrát, kolikrát se zvýší počet vstupů. [40]

Druhou třídou jsou pak problémy řešitelné (ověřitelné) v polynomiálně omezeném čase, avšak nedeterministickým polynomiálním algoritmem (NP). NP algoritmy s časovou složitostí např. $O(2^k)$ nebo $O(n!)$ pak mají dvě fáze:

- nedeterministická část odhadu správného (nejlépe optimálního) řešení,
- deterministická část testování, kde se ověřuje, zda je navržené řešení reálné dle daných omezení a kde se vyčísluje hodnota účelové funkce (kritérium optimality).

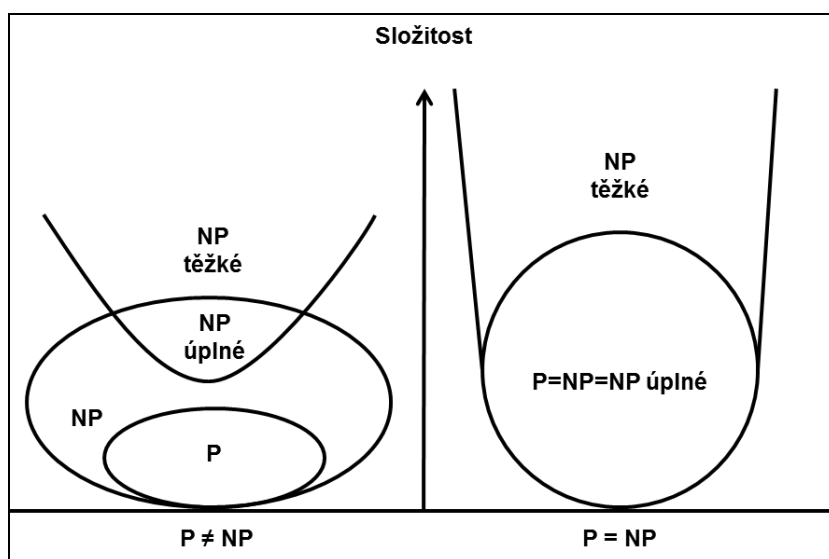
Takové algoritmy nedosáhnou požadovaného řešení v polynomiálním čase, avšak jsou schopné ověřit navržené řešení v tomto čase.

Třída NP problémů se pak dále dělí na:

- NP-těžké – existuje nedeterministický algoritmus, který ověří řešení problému v polynomiálním čase.
- NP-úplné - jsou takové nedeterministicky polynomiální problémy, na které jsou polynomiálně redukovatelné všechny ostatní problémy z NP.

Všechny NP-úplné problémy jsou NP těžké, ale některé NP těžké problémy nejsou NP – úplné.

Dodnes nevyřešenou otázkou pak zůstává, zda třída problémů $P=NP$ či $P \neq NP$ (Obr 3-1). To znamená, zda problémy, které lze ověřit v polynomiálním čase lze také v polynomiálním čase vyřešit. V tomto ohledu je publikována práce [41], která přislubuje vyřešení tohoto problému s tím, že $P \neq NP$. Závěry této práce jsou však brzy popřeny [42, 43].



Obr. 3-1: Eulerův diagram pro P, NP, NP těžké a úplné problémy

V oblasti plánování a rozvrhování výroby lze pokládat za P problém hromadění identických výrobků vyráběných pro různé výrobní zakázky do jedné či více výrobních dávek (tzv. neadresné plánování). V tomto případě algoritmus vyhledá ve fixovaných výrobních úkolech výskyt neadresně vyráběných dílců a na základě jejich četnosti určí počet výrobních dávek dle optimální či minimální výrobní dávky.

Za NP problém lze pak pokládat určení pořadí výrobních zakázek (např. ABCDEF) tak, aby byla celková doba výroby co nejkratší. V takovém případě se navrhne řešení (např. ACBDFE), které je pak následně rozvrhovacím algoritmem

ověřeno. Tento problém lze samozřejmě vyřešit např. úplnou enumerací, což je však časově velmi náročné $O(n!)$ (v tomto případě $6! = 720$ možných rozvrhů).

3.2 Rozvrhovací modely optimalizačních problémů

Teorie rozvrhování výroby v současné době čítá nezměrné množství publikací optimalizačních problémů [25, 29]. Řada z nich se zaměřuje na klasické teoretické problémy, jiné na jejich modifikace na konkrétní problémy. V rozvrhování výroby se pak nejčastěji mluví o m strojích M_j ($j=1, \dots, m$), které musí zpracovat n zakázek J_i ($i=1, \dots, n$). Rozvržením zakázek se rozumí alokace zakázky na jeden či více strojů v jednom či více intervalech. Jestliže zakázka J vyžaduje pro rozvržení více o operací O_k ($k=1, \dots, o$), pak $O_k(i, j)$ zpravidla odkazuje na rozvrhovací krok či operaci zakázky j na stroji i . Procesní čas p_{ij} spolu s termínem zahájení operace e_{ij} určuje dobu dokončení f_{ij} .

Pro popsání následujících modelů je nutné dále zavést následující notace modelů ve formě $\alpha|\beta|\gamma$, kde α označuje základní trajektorii zakázky přes stroje, β určuje charakteristiky zpracování a dalších omezení a γ specifikuje optimalizační kritérium (více viz kapitola 4.1).

Proměnná α určující druh modelu nabývá hodnot u [29]:

- Otevřený problém (z angl. open shop) $\alpha=O$
- Proudový problém (z angl. flow shop) $\alpha=F$
- Sekvenční problém (volně z angl. job shop) $\alpha=J$
- Smíšený problém (z angl. Mixed shop scheduling problem) $\alpha=X$
- Více komponentní rozvrhovací problém (z angl. Multi-component scheduling problem). $\alpha=MC$

Za proměnnou modelu (α) se dle počtu strojů zpravidla uvádí počty strojů daného modelu (např. v případě že $m=2$ tak $O2$). U klasických problémů se hovoří zejména o tzv. čtvercových problémech kde $m=n$.

V případech, kdy se jedná o modifikace problémů např. flexibilní, je notace před základním modelem (např. flexibilní proudový problém $\alpha=FF$).

Rozvrhovací problémy bývají v některých případech kombinovány s problémy rozvrhování pracovníků (např. roboticky ovládaný proudový problém $\alpha=F2, R1$ jsou dvě pracoviště ovládané jedním robotem).

Proměnná β určující další omezení modelu nabývá hodnot [29]:

- V případě, že termín zahájení alespoň jedné zakázky nelze realizovat v nulový čas ($e_{0j} \neq 0$), tzn. žádná z operací nemůže začít dříve než v čase r_i .

$$\beta = |r_i|$$

- Jestliže lze přerušit jakoukoli operaci pro uvolnění stroje pro jinou s následným dokončením později.

$$\beta = |prmp| \quad (\text{z angl. Preemption – dát přednost místo jiné})$$

- Jestliže operace prochází stroji v pořadí daném na prvním stroji (jedná se zejména o omezení pro $\alpha=F$).

$$\beta = |prmtn| \quad (\text{z angl. Permutation})$$

- Jestliže provedení operace je závislé na operaci z jiné zakázky (používá se v případech např. lakoven kdy je touto vazbou určeno pořadí barev).

$$\beta = |tree| \text{ (někdy také } \beta = |prec| \text{ z angl. Precedence)}$$

- Jestliže je procesní čas p_{ij} roven jedné časové simulační jednotce

$$\beta = |p_{ij}=1|$$

- Jestliže počet operací na stroji je např. menší nebo roven dvěma

$$\beta = |n_j \leq 2|$$

- Jestliže následující operace musí začít bezprostředně po ukončení předcházejících

$$\beta = |nowait|$$

- V případech závislosti seřizovacího času na sekvenci zakázek

$$\beta = |S_{sd}|$$

- Jestliže zakázka navštíví stroj či středisko více než jednou

$$\beta = |recrc|$$

- V případě, že jsou stroje či střediska blokována nedostatečnou kapacitou pro provedení následující operace

$$\beta = |block|$$

- Jestliže lze zkrátit čas trvání operace za určitou cenu

$$\beta = |contr|$$

V případě účelové funkce proměnná γ pak nabývá hodnot[29]:

- Maximální čas dokončení všech zakázek (makespan)

$$\gamma = C_{max}$$

- Celkový čas všech zakázek (total flow time)

$$\gamma = \sum C_i$$

- Maximální zpoždění všech zakázek (lateness)

$$\gamma = L_{max}$$

- Celkové odchylka od termínu všech zakázek (tardiness)

$$\gamma = \sum T_i$$

- Celkové penále za nedodržení požadavku s ohledem na zakázku

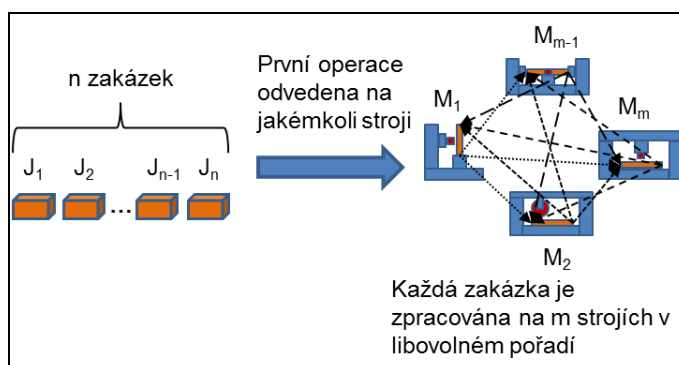
$$\gamma = \sum U_i$$

- Modely, kde $\beta = |contr|$, přičemž se snažíme o dosažení co nejmenšího C_{max} s nepřesáhnutím maximálně možné ceny za zkrácení K či minimální ceny K za zkrácení procesního času přičemž C_{max} nesmí přesáhnout daný C (pak je další podmínkou $\beta = |C_{max} \leq C|$)

$$\gamma = K \text{ či } \gamma = K, C_{max}$$

3.2.1 Otevřený problém (Open Shop Scheduling Problem)

Otevřený problém (OSSP) je speciálním případem rozvrhovacího problému, ve kterém pořadí operací na zakázce není předem určeno (obr. 3-2).



Obr. 3-2: Diagram obecného otevřeného problému (OSSP)

V případě klasického problému je počet operací roven počtu strojů. V některých případech je však procesní čas operace nulový.

V oblasti rozvrhování OSSP se vyskytují úlohy, které lze vyřešit polynomiálním algoritmem. Většina algoritmů, které řeší reálné problémy, je však označováno jako NP-těžké.

Nejtriviálnější úlohou je polynomiálně řešitelný $O2||C_{max}$, jehož složitost algoritmu je $O(n)$ [44]. Pro tento případ lze využít i prioritního pravidla LAPT (Longest Alternative Processing Time), které vybírá na volném stroji operaci s nejdelším procesním časem [29]. Toto prioritní pravidlo lze využít pro nalezení optimálního C_{max} v $O(n)$ pro $O2|prmp|C_{max}$. Více o prioritních pravidlech viz kapitola 4.4.1.1.

Mezi další klasické problémy řešitelné v polynomiálním čase patří například $Om|prmp|C_{max}$, řešitelný $O(r^2)$ algoritmem, kde r je počet operací s nenulovým procesním časem či $Om|prmp, r_i|C_{max}$ pro jehož řešení lze použít algoritmus se složitostí $O(pn)$, kde p je počet termínů uvolňování zakázek do výroby [45].

V případě, že tyto problémy rozšíříme na $Om||C_{max}$, kde $m > 2$, je složitost algoritmu pro řešení NP-těžká [44]. Složitost algoritmu však neovlivňuje jen množství strojů resp. operací, avšak i kritériální funkce. U problému $O||L_{max}$ nedosáhneme optimálního řešení v polynomiálním čase, tento problém je NP-těžký [45].

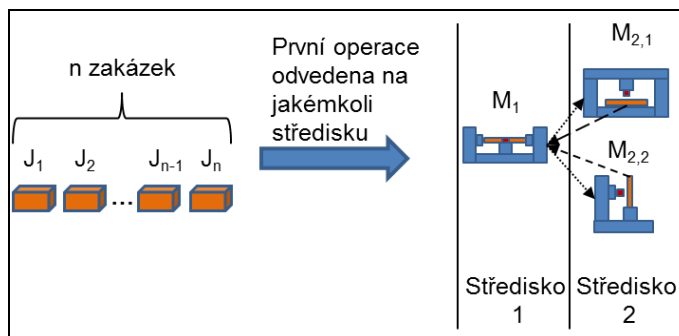
Omezující podmínky mají také zásadní vliv. Problém $O2|r_j|C_{max}$, který je redukovatelný na $O2||L_{max}$ je také NP- těžký.

Mezi problémy, jejichž aplikace na rozdíl od klasických problémů je častější patří problémy $O2|contr, C_{max} \leq C|K$, řešitelné v $O(n)$, a $O2|contr|K, C_{max}$ řešitelné v $O(n \log n)$ [46].

Jedná se o typické případy středních a malých podniků, které se zaměřují na servisní činnosti. Hlavním úkolem je redukovat procesní či seřizovací čas při dodržení či snaze o maximální přiblížení se k danému zákaznického termínu.

Další modifikací OSSP je flexibilní otevřený problém. Flexibilita u rozvrhovacích modelů se projevuje možností zařadit danou operaci na středisko. Například u $FO2|prmp|C_{max}$ (řešitelné v $O(n)$) to znamená, že každá zakázka je

zpracována na dvou střediscích, kde na prvním může být jeden stroj a na druhém stroje dva (obr.3-3), které jsou schopné zpracovat požadovanou operaci buďto za stejnou nebo rozdílnou časovou jednotku[46].



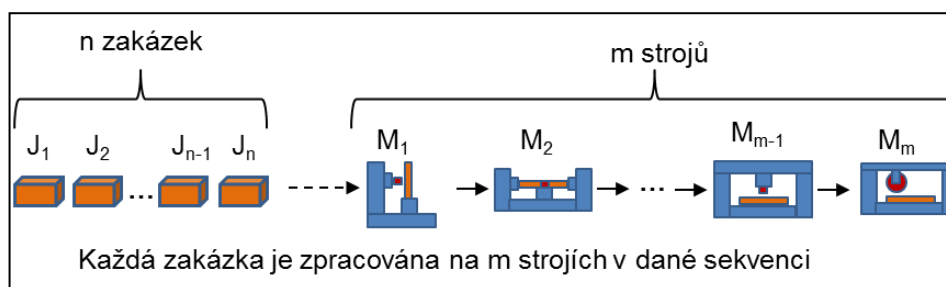
Obr. 3-3: Flexibilní model otevřeného rozvrhovacího problému (FO2)

Jedním z často používaných modelů, zvláště v opakované výrobě s vysokým prvkem automatizace, jsou tzv. cyklické otevřené problémy $O_{cm}||C_{max}$ (pro $m>2$ se jedná o NP-těžký problém) [47]. V těchto problémech se vytvořené rozvrhy opakují v cyklech s minimalizací celkových prostojů. Hlavním příkladem mohou být tzv. kompaktní cyklické problémy $O_{ccm}||C_{max}$, (pro $m>2$ se jedná o NP-těžký problém) typické pro chemický či hutní průmysl, pro které jsou generovány pouze rozvrhy bez zpoždění (více viz kapitola 4.2). V takových modelech je snaha o neustálé udržování cyklu výroby bez přerušení, zejména vzhledem k probíhajícím fyzikálně-chemickým procesům.

Lze tedy konstatovat, že až na výjimky jsou OSSP modely pro $m>2$ NP-těžké problémy, jejichž složitost může také ovlivnit kritériální funkce či další omezení výroby. V praxi se vzhledem k výše popsanému vyskytují zvláště problémy NP-těžké a to zejména v oblasti servisu. Příkladem může být organizace složená z mnoha středisek poskytující servis dopravního letadla. U takovéto zakázky se mohou vyskytnout dva rozdílné a nezávislé úkoly jako je oprava motoru a elektrických rozvodů. V takovém případě nezáleží na pořadí prováděné údržby, avšak obě operace nelze provádět najednou. Obdobné příklady lze nalézt v automobilovém průmyslu, v centrech kontroly kvality, v oblasti vychystávání do specializovaných obalů (tzv. bonboniéry), ve výrobě polovodičů či v oblasti satelitní komunikace [48].

3.2.2 Proudový problém (Flow Shop Scheduling Problem)

Proudový problém (FSSP) je takovým rozvrhovacím problémem, kde pořadí operací na různých zakázkách je stejné (obr. 3-4).



Obr. 3-4: Model proudového rozvrhovacího problému (FSSP)

Jeden z prvních článků o FSSP řeší $F2||C_{max}$ pomocí $O(n \log(n))$ (jedná se tedy o P problém) Johnsonova algoritmu[29], který produkuje tzv. $\beta = |prmtn|$ rozvrhy.

Parametry ovlivňující složitost algoritmu ovlivňuje stejně jako u OSSP počet strojů. $Fm||C_{max}$, stejně jako u OSSP je pro $m > 2$ NP-těžký problém [49].

Dalším z parametrů, který ovlivňuje složitost je čas uvolnění zakázky do výroby, $F2|r_i|C_{max}$ a $F2|prmtn, r_i|C_{max}$ také patří mezi NP-těžký problém.

Parametr, který se častěji objevuje u FSSP než u jiných rozvrhovacích problémů, je technologická závislost operace na operacích v jiných zakázkách. Do této oblasti patří např. $F2|tree, p_{ij}=1|C_{max}$ a $F2|tree, p_{ij}=1|\Sigma C_i$, které patří mezi P problémy a $Fm|tree|C_{max}$ popř. $Fm|tree|\Sigma C_i$ pro $m \geq 2$, které jsou jedny z nejobtížnějších NP-těžkých problémů.

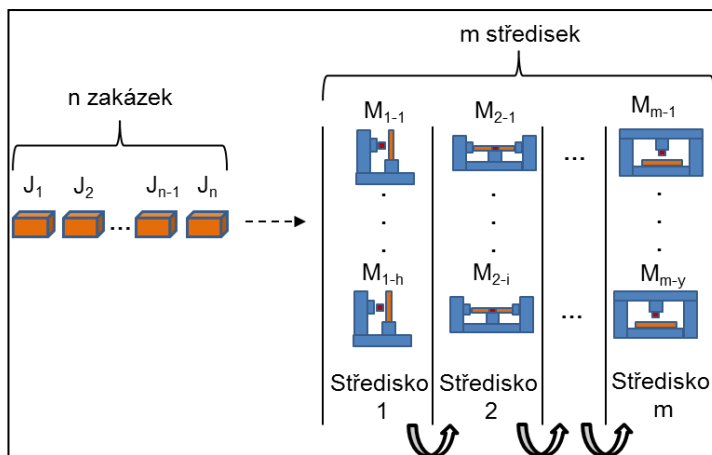
Velmi častým omezením vyskytujícím se v reálné výrobě je požadavek na bezprostřední zpracování následující operace. Příkladem může být zpracování ingotů v rámci technologického procesu kování či v případě tažení profilů za horka. Model $F2|nowait|C_{max}$ je řešitelný $O(n^2)$ algoritmem (tedy P problém), avšak modely jako jsou $Fm|nowait|L_{max}$ nebo $Fm|nowait|\Sigma C_i$ pro $m \geq 2$ jsou NP-těžké problémy.

V poslední době se FSSP spojují s omezením obslužnosti a rozvrhováním obsluhy ať již lidskými operátory či robotickými pracovišti a zpožděními způsobenými transportem. [50]. Již minimální modely $F2, R1|p_{1,j}=p_{2,j}=p_j; t_j \in \{T1, T2\}|C_{max}$, kde $t_j \in \{T1, T2\}$ jsou intervaly dopravních zpoždění, je NP-těžký.

Typ optimalizační (kriteriální) funkce také silně ovlivňuje složitost řešeného problému. Stejně jako tomu bylo u OSSP. Modely $F2||L_{max}$ a $F2||\Sigma C_i$ patří mezi NP-těžké.

Mezi nejčastější aplikace FSSP patří flexibilní (v některých případech označovány jako hybridní) modely, jejichž složitost bývá v naprosté většině případů NP-těžká [51].

Úkolem rozvrhování FFSSP je umístit zakázky v sekvenci na střediska zpravidla technologicky stejných strojů. Obrázek 3-5 zobrazuje model tohoto problému, kde je několik strojů využíváno ke zpracování jedné operace tak, aby pokryly přetížení úzkého místa. Stroje jsou označeny M_{x-y} , kde x určuje číslo střediska a y pak číslo stroje na daném středisku. M_{1-h} je tedy h stroj prvního střediska.



Obr. 3-5: Model flexibilního proudového problému (FFSSP)

Obtížnost řešení tohoto problému spočívá v určení pořadí zakázek na vstupu do výrobního systému, ale také na určení stroje na středisku, který danou operaci zpracuje. Přestože existují speciální případy kdy je FFSSP je P problém [52], již základní $FF2||C_{max}$ [53] je NP-těžkým problémem.

Stejně jako u předešlých modelů je tento problém rozsáhle modifikován zvláště s ohledem na praktická omezení. Nejčastěji se vyskytují modifikace jako závislost seřazení na sekvenci zakázek [54, 55, 56, 57, 58, 59], jejich přerušování [52], navracení na již použitá střediska [60], či nedostupnost strojů [61].

Tento druh modelu je v literatuře i v praxi řešen velmi obsáhle díky jeho aplikovatelnosti a to i co se týče jednoduchých dvou-střediskových modelů. Příkladem může být v elektrotechnickém průmyslu [54, 55] oblast kontroly křemíkových desek, kde probíhá optická a teplotní kontrola. V tomto případě se vhodnou volbou pořadí zakázek minimalizuje čas na seřazení (výměna a kalibrace sondy robotického pracoviště).

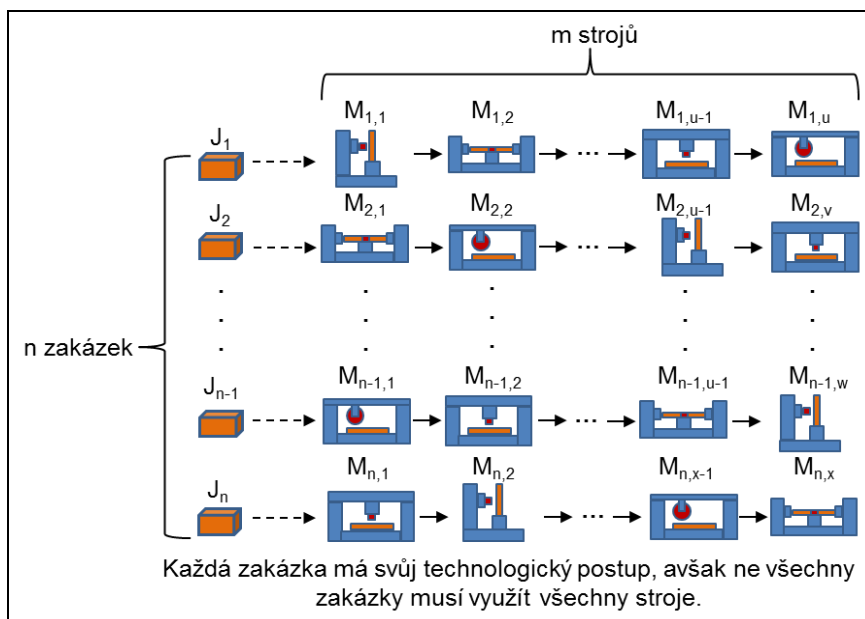
Dalším příkladem může být papírenský průmysl se dvěma středisky tiskem a povrchovou úpravou kalcitem či uhličitanem vápenatým [56], či v textilním průmyslu se středisky řezání stříhu a šití textilu [57].

Mezi složitější (s větším počtem středisek) příklady použití patří například výroba betonu se středisky míchání směsi, tvrzení směsi tlakem, řezání na požadované tvary a finální tvrzení za vysokých teplot.

Jedny z neobtížnějších modelů FFSSP se vyskytují v automobilovém průmyslu v oblasti výroby a montáže karoserií. Na jednotlivých střediscích se řeší obvykle rozdílná omezení. Na středisku lisování se jedná o určení pořadí jednotlivých dílů tak, aby sekvence co nejvíce minimalizovala čas potřebný pro přípravu formy. Na svařovacím středisku pak bývá problémem seřizování svařovacích robotických pracovišť na rozdílnou tloušťku materiálu, optimalizace svařovací dráhy a hlavně určování sekvence tak, aby byly všechny díly dostupné na kompletaci označované jako „svatba“. Středisko lakovny pak často využívá omezení řazení zakázek dle barevných škál od nejsvětlejší po netmavší. To vše musí pak v konečném důsledku odpovídat sekvenci dané předpovědi a zákaznickým požadavkům ve finální zástavbě karoserie [58].

3.2.3 Sekvenční problém (Job Shop Scheduling Problem)

Klasický sekvenční problém je rozšířením proudového problému [25] a je považován za nejobecnější model pro rozvrhování výroby [62]. Rozšíření spočívá v libovolné trajektorii výrobních zakázek, oproti proudovému problému, kde všechny zakázky mají stejnou trajektorii. Obrázek 3-6 pak zobrazuje model sekvenčního problému, kde označení strojů $M_{n,m}$, kde n udává číslo zakázky a m pak pořadí operace na dané zakázce.



Obr. 3-6: Model sekvenčního rozvrhovacího problému (JSSP)

Úkolem je pak rozvrhnout n zakázek na m strojích tak, aby se minimalizovalo či maximalizovalo dané kritérium. V oblasti klasických problémů se pak jedná o rozvrhnutí nm operací (tzn. všechny zakázky mají stejný počet operací).

V oblasti řešení sekvenčního problému existuje jen velmi málo případů, které by byly řešitelné P algoritmem. Mezi ně patří $J2|n_i \leq 2|C_{max}$ řešitelný redukcí na $F2||C_{max}$ řešitelný pomocí $O(n)$ [25] či $J2|p_{ij}=1, r_j|C_{max}$ a $J2|p_{ij}=1|L_{max}$ řešitelný v $O(n^2)$. [63]. Problém $J2|p_{ij}=1|C_{max}$ lze pak řešit $O(n \log n)$ algoritmem [64] a $J2|p_{ij}=1|\sum U_i$ algoritmem $O(n^6)$. Mezi problémy s rozšířeným omezením řešitelných P algoritmem patří $J2|n_i=1, pmtn|C_{max}$.

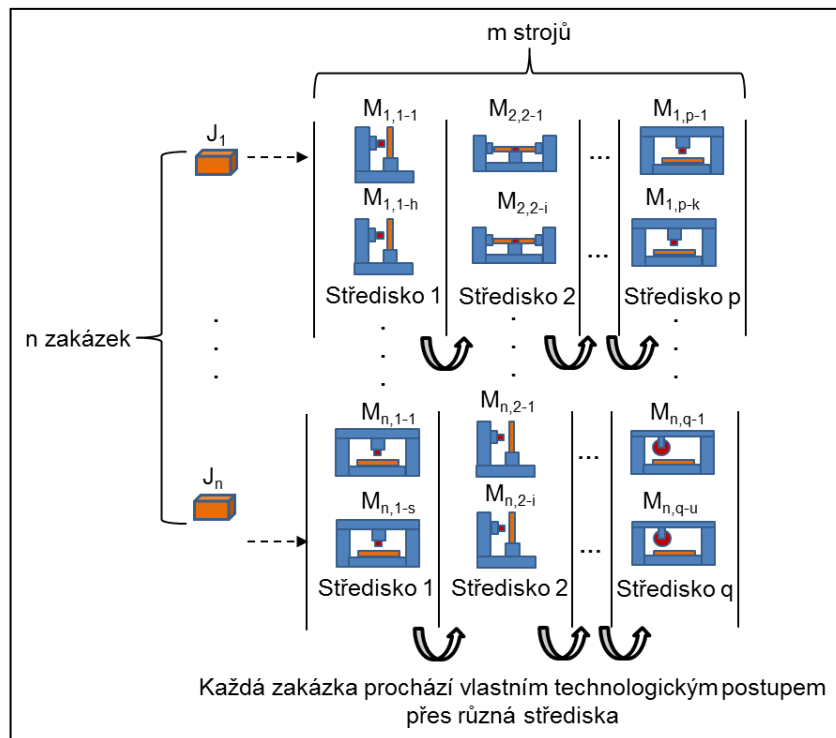
O složitosti tohoto rozvrhovacího problému vypovídá, že všechny $\alpha=Jm$, kde $m \geq 3$ včetně několika $m \leq 2$ jsou řešitelné pouze NP algoritmem (např. $J2|p_{ij}=1, r_i|\sum C_i$ nebo $J2|p_{ij}=1|\sum T_i$).

Obtížnost řešení takového modelu bývá složitější oproti FSSP zvláště z důvodu nutnosti určení pořadí zakázek před každým ze strojů na rozdíl od FSSP, kde určujeme zpravidla pouze sekvenci na vstupu do modelu.

Častou modifikací zvláště ve strojírenské výrobě, kde je stále používané technologické uspořádání výroby, je flexibilní sekvenční problém FJSSP (z ang. Flexible Job Shop Scheduling Problem).

U tohoto problému se určuje pořadí zakázek na jednotlivých výrobních střediscích a navíc oproti klasickému JSSP je třeba určit přiřazení každé operace na daný stroj. Tento problém je tedy rozšířen o problém přiřazovací. JSSP o popisované komplexitě je řešitelný pouze pomocí NP algoritmu [65].

Obrázek 3-7 zobrazuje model flexibilního rozšíření, kde jsou stroje označovány $M_{n,y-z}$, kde n určuje číslo zakázky $p(q)$ středisko a u stroj na daném středisku, kde se zakázka zpracovává.



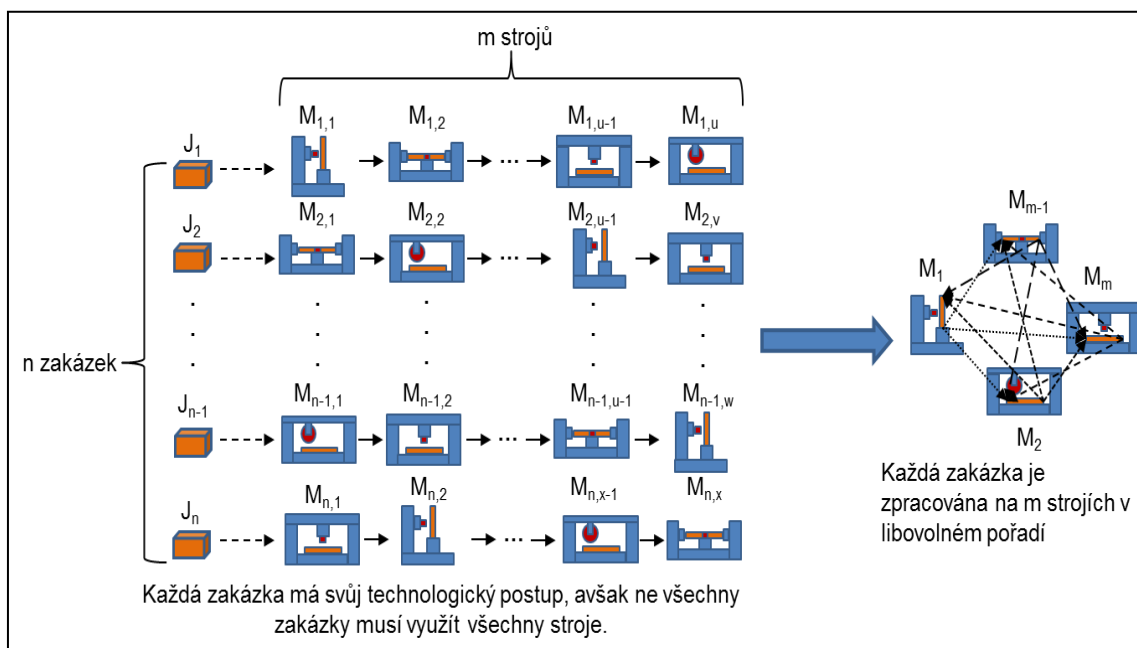
Obr. 3-7: Model flexibilního sekvenčního problému (FJSSP)

Sekvenční problém a jeho flexibilní modifikace nalézají řadu uplatnění jak v transportu [66], tak výrobě polovodičů [67], výrobě betonových prefabrikátů [68], rychlostních retardérů [69]. Aplikace tohoto modelu nachází uplatnění také v automobilovém průmyslu [70] či výrobě a servisu leteckých motorů [71].

Mezi známé aplikace patří např. výroba strojů na výrobu cigaret a balících strojů [72]. Flexibilní model obsahuje jak jednoúčelové frézky, tak CNC soustružnicko obráběcí centra a 3 až 5-osé brusky a soustruhy soustředěných v technologických centrech. Jednotlivé operace lze vyrábět na libovolném stroji na centru s tím, že některé operace lze z kapacitních důvodů přearadit na jednoúčelové stroje.

3.2.4 Smíšený problém (Mixed Shop Scheduling Problem či Group Shop Scheduling Problem)

Smíšený rozvrhovací problém (či skupinový rozvrhovací problém) je kombinací základních dvou rozvrhovacích problémů a to JSSP a OSSP. Úkolem je rozvrhnout všechny zakázky, z nichž část má předem danou unikátní trajektorii výrobním systémem (JSSP) a část zakázek je rozvržena bez omezení pořadí prováděných operací. Obrázek 3-8 znázorňuje systém, kde v první části se zpracovávají operace v např. technologicky uspořádané výrobě v prostředí JSSP a následně dochází k montáži, která není podmíněna sekvencí danou technologickým postupem.



Obr. 3-8: Smíšený rozvrhovací problém (XSSP)

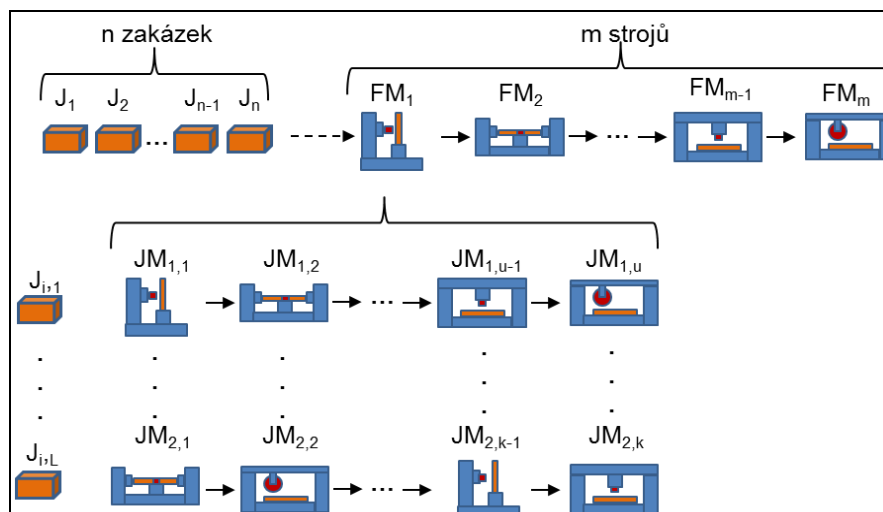
Přesto, že se jedná o kombinaci dvou často užívaných modelů, málo početně dostupná literatura [73,74] se věnuje většinou pouze komplexitě základních modelů, jejichž řešení lze nalézt pomocí P algoritmu. V oblasti smíšeného rozvrhovacího problému jsou známy pouze dva příklady modelů řešitelných pomocí P a to $X2|n_j \leq 2|C_{max}$ a $X2|pmtn, n_j \leq 2|C_{max}$ v čase $O(n \log n)$. Dále je znám model $X2|n_j \leq 2, pmtn|C_{max}$, kde n_j je počet operací v modelu JSSP[75]. O komplexitě tohoto problému vypovídá, že již $X|n=2|C_{max}$ a $X|n=2|\Sigma C_i$ jsou NP-těžké problémy [76].

Použití tohoto modelu se se zpravidla omezuje na oddělené teoretické JSSP a OSSP problémy s předpokladem využití v praktických oblastech využití těchto modelů [77, 78, 79].

3.2.5 Více komponentní rozvrhovací problém (Multi-component Scheduling Problem)

Více komponentní rozvrhovací problém resp. jeho model nebývá často popisován v technické literatuře a je často pojmenováván jako skupinový[80] rozvrhovací problém a v některých výjimečných případech jako smíšený rozvrhovací problém vzhledem k tomu, že jeho principy jsou tomuto modelu blízké. Zatímco smíšený rozvrhovací problém slučoval rozvrhovací a otevřený problém, více komponentní slučuje všechny základní modely (OSSP, FSSP a JSSP). Příkladem takového modelu je výrobní systém, kde zakázka ideologicky využívá [80, 81] proudový model a jednotlivé podčásti mají výrobní trajektorii totožnou s rozvrhovacím problémem.

Na obrázku 3-9 je zobrazen model, kde FM_m poukazuje na fázi trajektorie v rámci proudového principu rozvrhování a $JM_{x,y}$ na fázi modelu sekvenčního rozvrhovacího problému. $FM_4-JM_{2,3}$ tedy značí třetí stroj druhé zakázky v rámci čtvrté fáze celého výrobního procesu.



Obr. 3-9: Model více komponentního rozvrhovacího modelu (MCSSP)

Příkladem takového modelu může být oprava televizorů. Každý televizor má stejnou procesní trajektorii výrobním systémem a to přes pracoviště analýzy, opravy nebo výměny vadného komponentu a testování. Jedná se tedy o proudový systém. Každý z procesů, zejména proces opravy, může probíhat na různých strojích v různém pořadí zakázku od zakázky.

Literatura popisující tento model se komplexitou nezabývá vzhledem k tomu, že pro takový to systém se uvažuje minimální model $F3||C_{max}$ spolu s $J3||C_{max}$ s tím, že jakýkoli model ať již FSSP či JSSP pro $m \geq 3$ je NP-těžký.

Konstrukce řešení takového problému v rámci polynomiální fáze algoritmu se provádí pomocí semi-aktivního, aktivního rozvrhu či rozvrhu bez zpoždění (více viz. kapitola 4.2) a tedy podobným způsobem, jak je tomu u JSSP [25, 81].

3.3 Dílčí závěr č.1 - Výběr modelu pro co nejširší uplatnění ve strojírenské výrobě

Cílem předešlých kapitol bylo analyzovat modely resp. optimalizační problémy z oblasti rozvrhování výroby z hlediska složitosti nalezení řešení, z pohledu vlastností materiálového toku a možností nasazení v průmyslové praxi. V této části je provedeno shrnutí s ohledem na dílčí cíl 1, tedy navržení vhodného modelu pro použití v středních a malých podnicích v oblasti strojírenské výroby.

3.3.1 Složitost optimalizačních problémů

Složitost optimalizovaných modelů resp. možnost nalezení přesného řešení v čase je u všech modelů závislá na druhu okrajových podmínek (např. $F2|tree,p_{ij}=1|\Sigma C_i$, které patří mezi P problémy a $F2|tree|C_{max}$ popř. $F2|tree|\Sigma C_i$ patřící mezi NP problémy).

Dalším prvkem ovlivňujícím složitost problému je účelová funkce. Příkladem může být porovnání modelů $F2|nowait|C_{max}$ patřících mezi P problémy a ekvivalentních modelů lišících se v účelové funkci $F2|nowait|L_{max}$ nebo $F2|nowait|\Sigma C_i$, jenž jsou problémy NP-těžké.

Zásadním činitelem ovlivňujícím složitost řešeného problému je pak počet strojů a zakázek. Lze říci, že v naprosté většině případů modely obsahující více než 3 stroje a k nim příslušný počet zakázek (u teoretických problémů často $m=n$) jsou NP-těžké.

3.3.2 Využití modelu vzhledem k druhu materiálového toku

Z pohledu složitosti materiálového toku lze modely klasifikovat na základní tři. Základní výhodou otevřeného problému (OSSP) je možnost rozvrhovat operace bez omezené návaznosti na operace předešlé. Tato vlastnost se často v praxi vyskytuje v modelech, které jsou na první pohled klasickým proudovým či sekvenčním problémem. Dílenští plánovači v takových případech reagují intuitivně při volbě operace, avšak z hlediska požadavku na jednoznačnost materiálového toku se tomu předchází často přesným definováním pořadí operací v technologickém postupu, který je hlavním zdrojovým dokumentem pro plánování výroby. Problém se pak redukuje z klasického OSSP, kde určujeme pořadí vykonávaných operací a jejich pořadí na výrobních prostředcích, na problém FSSP či JSSP.

Proudový model (FSSP) je díky snaze o co největší standardizaci asi nejrozšířenějším modelem. V tomto modelu mají všechny operace určený materiálový tok skrze výrobní systém a to ve stejném pořadí pro všechny zakázky. Jedná se tedy o modely využívané zvláště pro velkosériovou výrobu. Hlavním úkolem při řešení problému s takovýmto modelem je pak určení pořadí operací na vstupu do výrobního systému (nejčastěji linky). Modely blízké FSSP vyskytující se v praxi však často obsahují problém vrácení se zakázek na pracoviště předešlá, což je typickým znakem následujícího modelu.

Základní vlastností sekvenčního modelu (JSSP) je, že každá zakázka má (zvláště u teoretických modelů) svoji unikátní trajektorii výrobním systémem. V praxi to znamená, že sekvence operací se liší od zakázky k zakázce. Nejčastěji se však jedná o modely velmi blízké JSSP, u kterých však dochází, jak již bylo zmíněno, k vrácení zakázek na stroje, na kterých již byly zpracovávány. Hlavním úkolem při rozvrhování výroby je pak určení sekvence zakázek na každém pracovišti. Jedná se tedy o daleko složitější problém než u FSSP.

Mezi další modely vyskytující se v odborné literatuře jsou smíšený rozvrhovací problém (kombinující modely JSSP a OSSP) a vícekomponentní rozvrhovací problém (kombinace všech modelů předešlých), které však vycházejí ze základní problematiky řešené v modelu JSSP s tím, že jsou některé precedenční vazby (nutná návaznost operací vzhledem k technologickému postupu) odstraněny. Nejčastěji se však jedná o přímou návaznost modelů, kde se nejprve řeší problém pomocí jednoho modelu a výstupy z něj jsou pak použity v modelu následujícím.

3.3.3 Aplikace modelů v praxi

Vzhledem k nastíněné problematice složitosti trajektorie zakázky výrobním systémem se tato část zabývá aplikací klasických modelů OSSP, FSSP, JSSP a jejich flexibilními modifikacemi.

Model OSSP je z těchto tří modelů v praxi a odborné literatuře nejméně zmiňovaný. To je dáno jeho použitím zvláště v oblasti servisu a služeb. Model se často vyskytuje jako část systému využívající FSSP a JSSP v případech, kde lze zaměnit pořadí prováděných operací. Hlavním problémem je pak využití takového modelu

v praxi s ohledem na neexistenci přesných dat. Technologické postupy uvedené v databázi ERP často neumožňují různý sled operací, a v případech, kdy zkušení dílenští plánovači si možnost změny pořadí operací v technologickém postupu uvědomují a využívají, tím mohou narušit plán vycházející právě z technologických postupů. Lze tak říci, že jeho využití je vhodné spíše pro řešení dílčích problémů, než komplexních výrobních systémů. Otevřený rozvrhovací model se využívá zejména v oblastech, jako jsou služby, servis či logistika, zvláště v případech vychystávání (viz tab.3-1).

Model FSSP patří mezi nejpopulárnější modely používané v praxi i často diskutované v odborné literatuře. To je historicky dáno snahou o co maximální sjednocení projektu v opakovatelné procesy. Z toho vyplývá použití modelu FSSP zvláště v sériové výrobě, kde se trajektorie různých produktů z pohledu technologie příliš nemění. V oblasti plánování a řízení výroby se pak v naprosté většině případů používají štíhlé metody vycházející z prostředí automobilového průmyslu (viz tab.3-1). Filozofie JIS a JIT společně s metodou Kanban a prioritním pravidlem FIFO tak potlačují nutnost rozvrhování. Informace co vyrábět je dána samoregulačními principy metody Kanban a požadavek v jakém pořadí prioritním pravidlem FIFO.

Model JSSP se v odborné literatuře vyskytuje častěji než FSSP díky jeho lákavosti z hlediska složitosti modelu (klasický JSSP má $n!$ FSSP pouze $n!$). Jedná se o model využívaný zvláště v kusové - zakázkové výrobě. Proto bývá tento model v české literatuře často označován [82] jako model zakázkové výroby. Jeho vlastnosti se však vyskytují i v malosériové výrobě, která bývá často řízena podobným způsobem, jako je tomu u modelu FSSP. Striktní využití pravidla FIFO pak může citelně omezit možný potenciál rozvrhu dosáhnout lepšího řešení z pohledu optimalizačního kritéria (dodací lhůta, celkový čas odvedení všech zakázek atd. viz kapitola 4.1), jak dokazuje množství aplikací z různých oblastí strojírenské výroby uváděných v odborné literatuře (viz tab. 3-1).

Tab. 3-1: Přehled aplikací modelů OSSP, FSSP, JSSP a jejich flexibilních modifikací

Model	Typ výroby	Známé aplikace v praxi *
OSSP	Vybrané oblasti z sériové a kusové výroby	Servis motorů [47]
		Vychystávání do zákaznických obalů (bonbonier)
		Výroba polovodičů
		Satelitní komunikace [48]
FSSP	Sériová výroba	Kontrola křemíkových desek [54, 55]
		Výroba papíru [56]
		Automobilový průmysl (výroba, montáž) [58]
		Textilní průmysl [57]
JSSP	Malosériová a kusová výroba	Transportní systémy [66]
		Výroba polovodičů [67]
		Výroba betonových prefabrikátů [68]
		Rychlostní retardéry [69]
		Automobilový průmysl (výroba tvarové a technicky složitých součástek) [Job9]
		Výroba leteckých motorů [71]
		Výroba strojů a zařízení [72]
		Výroba silničních a stavebních strojů
		Výroba kolejových vozidel

* Pozn.: Aplikace z praxe vycházejí z uveřejněných vědeckých článků, konzultací ve výrobních podnicích a projektů řešených na TUL ve spolupráci s průmyslem.

3.3.4 Shrnutí

V oblasti složitosti modelů lze konstatovat, že při aplikaci je nutné hledat nástroj řešení v oblasti NP problémů bez rozdílu ke druhu modelu (více viz kapitola 4). To vzhledem k základnímu předpokladu, že většina reálných modelů obsahuje více než 3 stroje.

Z předchozí analýzy modelů výroby je zřejmé, že nelze striktně určit nejpoužívanější model ve strojírenské výrobě. Toto nelze definovat ani v průmyslovém odvětví (např. automobilový průmysl). Z analýzy modelů však vyplývá, že vzhledem k tomu, že otevřený rozvrhovací problém řeší pouze podčásti proudových či sekvenčních modelů a že v oblasti rozvrhování proudového modelu se s úspěchem používají metody Lean, má největší možnost uplatnění principů rozvrhování model sekvenční (JSSP) a jeho flexibilní modifikace (FJSSP). Sekvenční rozvrhovací model je také považován za nejobecnější model využitelný ve strojírenské výrobě[62].

Pro další výzkum v rámci této disertační práce, vzhledem k provedené analýze současného stavu, byl vybrán model sekvenčního rozvrhovacího problému a jeho flexibilní modifikace.

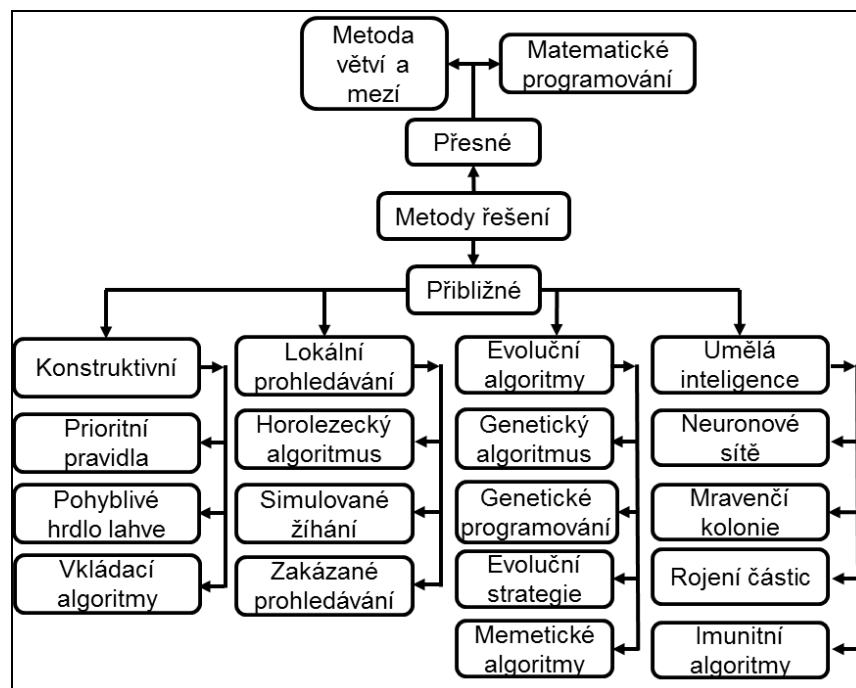
V následující kapitole jsou analyzovány metody pro tvorbu a optimalizaci dílenských rozvrhů.

4 METODY PRO TVORBU A OPTIMALIZACI DÍLENSKÝCH ROZVRHŮ

Přestože se problematikou rozvrhování operací zabývá odborná literatura již od počátku minulého století [83], žádný z výzkumů a aplikací nenalezl způsob, jak rychle nalézt optimální řešení, a to nejen teoretických modelů, ale ani praktických aplikací (vyjma problémů řešitelných v polynomiálním čase - jejich výskyt v praxi je však málo pravděpodobný).

Metody poskytující řešení JSSP problémů lze rozdělit na základní dva směry (obr 4-1). Jsou to takzvané metody přesné (někdy také nazývané optimalizační), které nacházejí přesné řešení např. enumerativním způsobem. To však má za následek dlouhý výpočetní čas. Aproximační (neboli přibližné metody) způsob je druhým směrem, který se již dlouhou dobu rozvíjí vzhledem k jeho větší možnosti aplikovatelnosti v praxi. Jejich aplikovatelnost je dána tím, že přestože nezaručují nalezení optimálního řešení, poskytují řešení v reálném čase.

Následující kapitoly se zabývají oběma směry, avšak vzhledem k předpokladu aplikovatelnosti v praxi je větší důraz kladen na metody aproximační.



Obr. 4-1: Metody pro tvorbu a optimalizaci dílenských rozvrhů

Před analýzou metod určených pro tvorbu řešení a jejich optimalizaci je nutné definovat kritérium hodnocení řešení – účelové funkce a pojem sousedství, který se vyskytuje u většiny následujících metod.

4.1 Účelová funkce

Účelová funkce, často nazývaná jako kritérium optimality, je měřítkem k porovnávání reálných rozvrhů (x). V kapitole 3.2 byly v krátkosti nastíněny základní optimalizační funkce, které se používají zvláště v oblasti teoretických rozvrhovacích problémů. Pro správné použití účelových funkcí je však důležité popsat jejich způsob

použití v praxi. V naprosté většině případů se pak jedná o minimalizaci takového kritéria.

Maximální čas dokončení všech zakázek (makespan)

Maximální čas dokončení všech zakázek neboli makespan (C_{max}) (1) [25] je nejpoužívanějším kritériem v oblasti teoretického výzkumu zvláště z důvodu nízké náročnosti na výpočet.

$$C_{max}(x) = \max_{1 \leq i \leq n} \{C_i\} \quad (1)$$

Jedná se maximální čas dohotovení poslední operace. V praxi je toto kritérium užíváno pouze na úrovni jednotlivých středisek a to zejména v prostředí, kde se plánuje výroba pomocí principu tahu v rámci řešení MRP. V takovém případě má dílenský mistr v plánu frontu operací, ze které odebírá úkoly pro jednotlivá pracoviště. Bývá obvyklé, že vzhledem k principu tlaku, je na daná pracoviště navrhnut plán převyšující kapacity daných strojů, tzn. dílenský mistr je přirozeným způsobem tlačěn k tomu, aby všechny zakázky odvedl v co nejkratším čase.

Při minimalizaci tohoto kritéria se často maximalizuje celkový průtok systémem (tedy na úzkém místě) a zajišťuje se rovnoměrné zatížení strojů.

Celkový čas všech zakázek (total flow time)

Celkový čas všech zakázek ($\sum C_i$) (2) je velmi blízký kritériu makespan [84]. Zde se kontroluje celkový čas C_i , po který jsou zakázky zpracovávány na zdrojích počínaje časem zahájení první (C_F) operace a časem poslední operace (C_L) konče. Toto kritérium nalézá uplatnění zvláště v oblasti flexibilních rozvrhovacích problémů, kde převedením operace z jednoho technologicky podobného zdroje na jiný lze zkrátit daný čas operace.

$$\sum C_i = \sum_{i=1}^n (C_{L(i)} - C_{F(i)} - p_{l(i)}) \quad (2)$$

Kde p_i je čas první operace, který stráví na zdroji.

To je v praktických podmínkách často prováděno intuitivně na základě zkušenosti dílenského mistra.

Obdobou této účelové funkce je pak vážený čas všech zakázek ($\sum C_i w_i$ kde w_i je váha zakázky či stroje), kde se uvažuje důležitost zakázky (např. s ohledem na zákazníka) či vytížení stroje (např. s ohledem na termín návratnosti investic na pořízení).

Odchylka od zákaznického termínu (lateness)

Odchylka od zákaznického termínu (L_{max}) (4) [85] je kritérium vycházející z předpokladu penalizace společnosti za odchylku od termínu dodání zakázek. Znamená to tedy penalizaci jak v případě brzkého, tak pozdního dodání (3).

$$L_i = C_{L(i)} - d_i \quad (3)$$

Kde d_i je termín odvedené zakázky.

$$L_{max}(x) = \max_{1 \leq i \leq n} \{L_i\} \quad (4)$$

Je používáno v celostním pohledu na rozvrhování výroby, kde není důležité co největší vytížení strojů a co nejrychlejší zpracování všech zakázek, ale dodržení zákaznického termínu.

ΣL_i (5) pak je typické pro společnosti využívající principů štíhlé výroby a zejména pak principů právě včas (JIT).

$$\sum L(x) = \sum_{i=1}^n \{C_{L(i)} - d_i\} \quad (5)$$

Zpoždění zakázek (tardiness)

Zpoždění zakázek na rozdíl od odchylky od termínu dodání klade důraz na požadavek zákazníka, a tedy nepovažuje dřívější dohotovení zakázky (resp. zásoby vázané ve skladu) jako penalizaci.

$$T_i = \max \{0, L_i\} = \text{zpoždění zakázky } J_i, \quad (6)$$

Stejně jako odchylka od zákaznického termínu je i tato veličina uvažovaná v celkové (ΣT_i) (7) a maximální hodnotě (T_{\max}) (8). Celkové zpoždění všech zakázek (ΣT_i) [86] je kritériem uvažujícím všechna zpoždění, nejen maximální.

$$\sum T(x) = \sum_{i=1}^n \{\max \{0, L_i\}\} \quad (7)$$

$$T_{\max}(X) = \max_{1 \leq i \leq n} \{ \max \{0, L_i\} \} \quad (8)$$

Na toto kritérium navazuje vážené zpoždění zakázek [87] ($\Sigma T_i w_i$ - kde w_i je váha zpoždění), kde je zohledněna kritičnost v rámci technologie (zpoždění na úzkém místě), nebo času dodání (priorita dle důležitosti zákazníka).

Celkový předběh a zpoždění všech zakázek (earliness, tardiness) (9)

Toto kritérium ($\Sigma T_i + \Sigma E_i$) [88], kde

$$E_i = \max \{0, -L_i\} \text{ předstih zakázky } J_i, \quad (9)$$

navazuje stejně jako předešlá dvě kritéria na principy štíhlé výroby s důrazem na plýtvání. Bývá často používáno v kombinaci s použitím vah jednotlivých zakázek, jak co se týče předstihu, tak i zpoždění. Bývá někdy označováno s ohledem na penalizační jednotky[25]:

$$\sum U(x) = \sum_{i=1}^n [w_{1i} E_i + w_{2i} T_i] \quad (10)$$

Používá se (10) v případech, kdy pro společnost není důležité pouze dodávat zákazníkovi včas, ale i eliminovat náklady spojené s předčasným dokončením zakázky.

Vnitřní kritéria optimality

Tato kritéria nejsou používána v teoretických problémech a nepoužívají se zpravidla napříč celým výrobním systémem. Přesto jsou hojně využívána na pracovištích se specifickými požadavky:

- Minimalizace času seřizování [88] – vychází z požadavku na minimální finanční investici do nevýrobních činností, které ale zpravidla bývají nedílnou součástí výroby. Jedná se pak zejména o minimalizaci času na úzkých místech.
- Minimalizace počtu pracovníků [89] – využívá se zejména v oblasti montážních linek.
- Maximální vytížení strojů [90] – principy společné s kritériem C_{max} , avšak soustředí se na pracoviště, kde je důraz kladen na návratnost investice.
- Minimalizace zásob [91] – souvisí s ΣC_i , avšak v tomto případě je kladen důraz na kapitál vázaný v zásobách.
- Maximální robustnost (proveditelnost) [90] – zohledňuje se počet operací, které je nutné znovu rozvrhnout v případě změn (nedodaný materiál, porucha stroje, výkyvy ve výkonosti pracovníků atd.). Toto kritérium je vzhledem ke způsobu výpočtu (založeném na predikci) málo používané. V teorii je známé hlavně v oblasti tzv. fuzzy rozvrhování. Hlavním problémem je neexistence dat zvláště ve výrobní oblasti, která by dostatečně popsala pravděpodobnost výskytu.

Vícekritériální hodnocení

Oblast vícekritériálních optimalizačních funkcí se zabývá hodnocením dvou a více kritérií, které v oblasti rozvrhování lze řešit zpravidla pouze kompromisně. To znamená, že nelze nalézt takové řešení, které by bylo optimální pro všechna kritéria.

Příkladem může být situace, kdy se dílenský mistr snaží optimalizovat výrobu tak, aby odvedl všechny zakázky z jeho středisek co nejdříve, aby uvolnil kapacity strojů. Vychází z fronty operací dané systémem plánování do neomezených kapacit (MRP). Změnou pořadí operací (či výrobních dávek) snižuje celkový čas, což má však za následek uspořádání některých zakázek a zpoždění zakázek jiných, což jde proti obvyklému požadavku společností na JIT.

Dalším typickým příkladem protichůdných požadavků je snaha po maximálním vytížení strojů a přitom o minimalizaci zásob.

V těchto případech obvykle nelze nalézt takové řešení, které je optimální ve všech ze sledovaných kritérií. Snahou vícekritériálního hodnocení je pak nalezení kompromisu mezi často protichůdnými kritérii. Tento kompromis je však obvykle lepší než optimum získané pouze pomocí jednoho z kritérií.

Nejjednodušším způsobem vícekritériálního hodnocení může být již zmiňované kritérium předběh a zpoždění všech zakázek, kde $\gamma = \Sigma T_i + \Sigma E_i$ (lineární optimalizační model).

Pokročilejším způsobem je pak tzv. agregační způsob, kde přiřazujeme daným kritériím váhy $\gamma = \Sigma T_i w_i + \Sigma E_i q_i$.

V oblasti rozvrhování se dlouhodobě diskutuje tzv. princip pareto optimality resp. pareto dominance [92], který vybírá řešení pomocí porovnání všech dostupných např. pomocí bodování hodnoty (11).

$$Hodnost(x_i) = 1 + p_i \quad (11)$$

kde x_i je řešení a p_i počet řešení, které mu dominují (mají lepší hodnotu kritériální funkce).

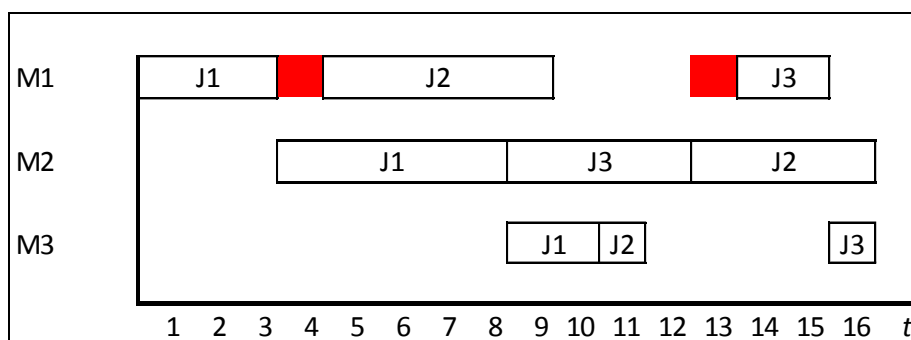
4.2 Množiny všech řešení a podmnožiny sousedních řešení

Množinu všech možných rozvrhů lze rozdělit na „neaktivní“, semi-aktivní, aktivní rozvrhy a rozvrhy bez zpoždění [29]. Rozdíly mezi těmito druhy rozvrhů lze popsat na následujícím příkladu modelu JSSP (tab. 4-1). Model tvoří tři zakázky (J1-J3), jež jsou zpracovávány na třech strojích dle technologického postupu (např. zakázka J1 nejdříve zpracovávána na stroji 1 s časovou náročností 3 časové jednotky).

Tab. 4-1: Model sekvenčního problému

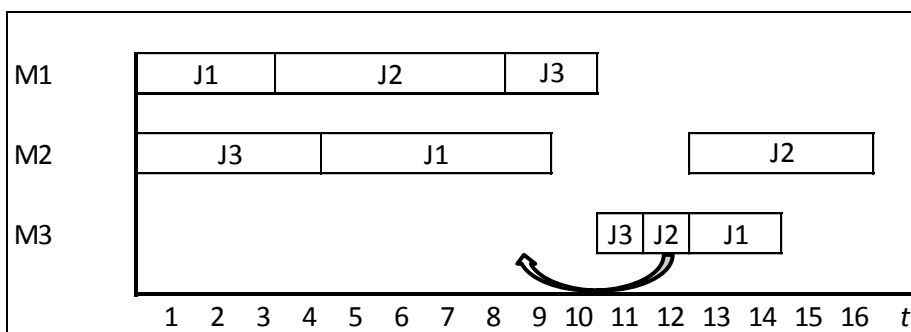
Zakázka	Technologický postup - stroj (procesní čas)		
J1	1 (3)	2 (5)	3 (2)
J2	1 (5)	3 (1)	2 (4)
J3	2 (4)	1 (2)	3 (1)

Za neaktivní rozvrhy lze považovat ty, u kterých lze pouhým posunem operace bez změny sekvence dosáhnout nižšího C_{max} (obr. 4-2).



Obr. 4-2: „Neaktivní“ rozvrh

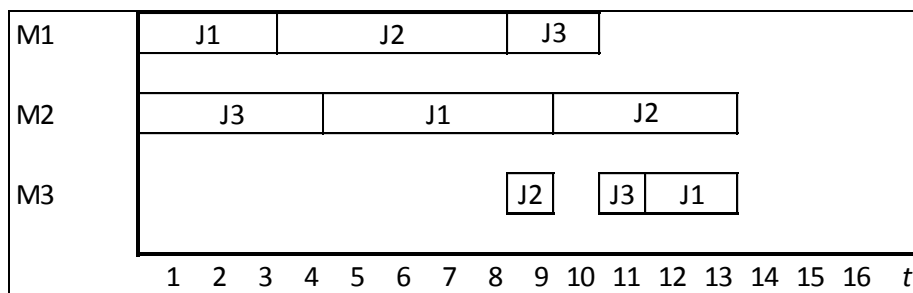
Semi - aktivní rozvrh je takový rozvrh, kde není možné zaplánovat operaci dříve, aniž by se změnilo pořadí, ve kterém vstupují zakázky na stroj (obr 4-3). Lze také říci, že je to takový rozvrh, kdy je operace rozvržena na nejbližší možný čas.



Obr. 4-3: Semi-aktivní rozvrh

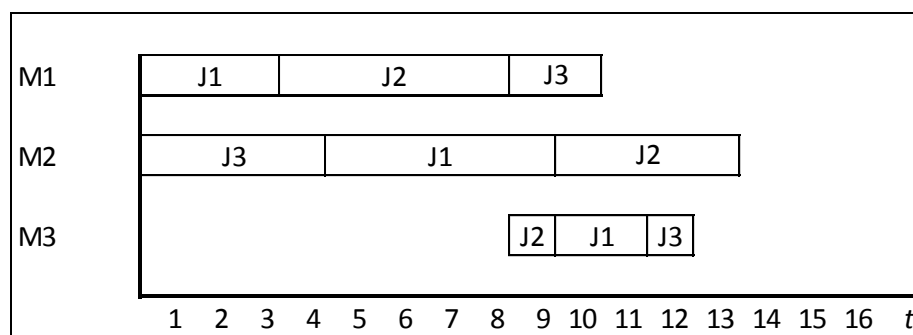
Aktivní rozvrh (obr. 4-4) je pak rozvrh, kde není možné vytvořit jiný rozvrh změnou sledu alespoň jedné operace na stroji tak, aby byla odvedena dříve, aniž by došlo k opoždění operace jiné [93]. Tento typ rozvrhu je nejčastěji používán při zkoumání optimality ze dvou důvodů. Prvním důvodem je, že optimální rozvrh je vždy

rozvrhem aktivním [93]. Druhým důvodem je velikost sousedství (okolí) aktivních rozvrhů, které bývá znatelně menší než u semi-aktivních (obr. 4-6).



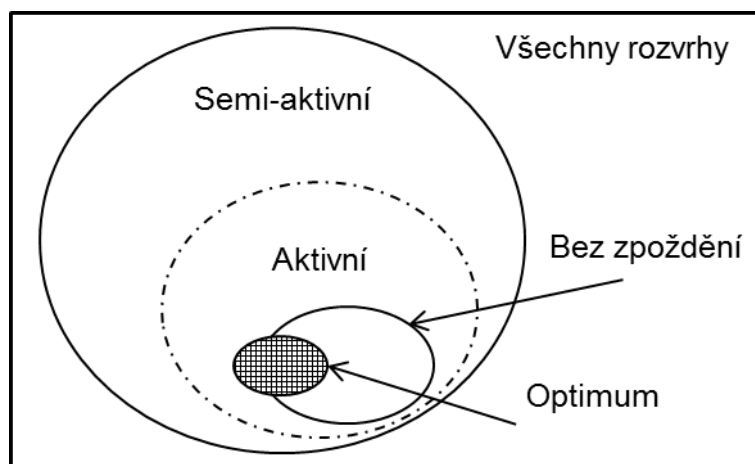
Obr. 4-4: Aktivní rozvrh

Posledním zmiňovaným rozvrhem je rozvrh bez zpoždění (ND - z anglického Non- delay schedule), který je podmnožinou rozvrhů aktivních. V takovém plánu není stroj bez práce (přiřazené operace), pokud je jakákoliv operace v daný čas pro tento stroj dostupná (obr. 4-5).



Obr. 4-5: Rozvrh bez zpoždění

V případě tohoto modelu, ačkoli se jedná o malý model, je celková množina všech rozvrhů $n=3!=261$.



Obr. 4-6: Mapa rozvrhů

Prohledávat zobrazené množiny řešení (obr. 4-6) pro nalezení toho optimálního lze pomocí hledání sousedních řešení. Dvě řešení se nalézají v sousedství, pokud jedno lze nalézt pomocí toho druhého vhodně nadefinovanou modifikací.

V rámci použití většiny optimalizačních metod v oblasti rozvrhování se definuje způsob generování sousedství, který pro každé reálné řešení x umožňuje pomocí množiny změn $S(x)$ určit okolí resp. množinu takto získaných řešení $N(x)$ (z angl. neighbourhood). Dalším důležitým prvkem při stanovení okolí (sousedství) je splnění podmínky, kdy každé řešení je dosažitelné z libovolného jiného řešení použitím $S(x)$.

Takovou změnou může být například prohození sousedních operací viz obrázky 4-4 a 4-5, kde došlo ke změně pořadí zakázek J1 a J3 na stroji M3 a tedy z rozvrhu (J1,J2,J3|J3,J1,J2|J2,J3,J1) na (J1,J2,J3|J3,J1,J2| J2,J1,J3) pro (M1|M2|M3).

4.3 Přesné metody řešení

Přesné metody v oblasti rozvrhování výrobních zakázek řeší dané problémy optimálně v polynomiálním čase, který obvykle roste exponenciálně s lineárním nárůstem problému. Tyto metody jednoduše vytváří optimální řešení z údajů o problému v návaznosti na jednoduchou sadu pravidel, které přesně určují pořadí zpracování. Prvním příkladem přesné metody a pravděpodobně prvním algoritmem zabývajícím se rozvrhováním výroby je Johnsonův algoritmus [94]. Ten se zabýval řešením rozvrhování proudové výroby libovolného počtu úloh na dvou strojích. Existují další metody přesného řešení JSSP, jako je grafické řešení rozvrhovacího problému [95] [62], které jsou však těžko aplikovatelné v dnešním prostředí. Práce se dále zabývá dvěma nejčastěji používanými metodami a to metodou větví a mezí, a metodou dynamického programování.

4.3.1 Metoda větví a mezí

Metoda větví a mezí je jedna z heuristických metod, která pracuje na základě rozdělování prohledávaného okolí [62]. Tato metoda je založena na ideji inteligentního enumerativního prohledávání všech řešení u kombinatorických problémů. Inteligentní znamená, že neprohledává slepě všechna řešení, ale pouze ta, která jsou reálná.

Pomocí této metody se stanoví důkaz, že řešení je reálné na základě úspěšného dělení prohledávaného místa a to dvěma fázemi větvením a stanovením mezí [96].

Větvením se rozumí kombinatorické dělení na základě rozhodovacích proměnných a omezení daných např. druhem rozvrhu (viz. kapitola 4.2). Při větvení se vychází z tzv. kořene stromu, ze kterého vychází varianty první proměnné. V případě rozvrhování se tedy jedná o větve prvních dostupných operací.

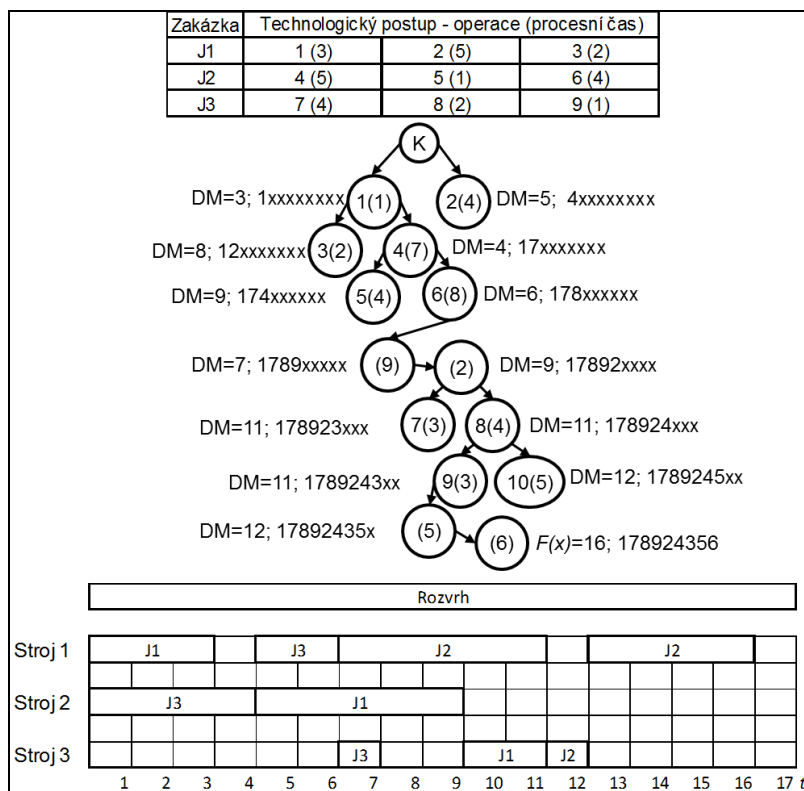
Ve fázi určování mezí se pak pro takto získané větve určí horní a dolní mez sledovaného parametru vzhledem k účelové funkci (např. u funkce makespan to je čas dokončení operace). Tento krok nejenže určuje výslednou hodnotu účelové funkce, ale také výrazně napomáhá ke zkrácení doby prohledávání (prohledávání s návratem). V případě, kdy bylo v předchozích krocích získáno řešení x_0 a jeho hodnota účelové funkce $f(x_0)$ a hledá se lepší řešení než je hodnota $f(x_0)$ v jiné větvi daného stromu, lze ukončit prohledávání dané větve v případě, že uzel nabývá hodnoty dolní meze horší než $f(x_0)$. Taková větev se označí za prohledanou a dále se nepokračuje v jejím větvení.

Výběr následujícího uzlu je řízen prohledávající strategií, např. prohledáváním do šířky nebo do hloubky [40, 97].

Princip prohledávání do šířky je založen na výběru uzlu s minimální dolní mezí ze všech neprozkoumaných uzlů. V takovém případě prohledávání může přeskakovat z větve na větev. Nevýhodou tohoto principu je velká šíře prohledávaného prostoru

způsobující dlouhý čas potřebný k dosažení jakéhokoli řešení. Kladem této strategie je nalezení kvalitnějšího počátečního řešení, které pak lze použít pro omezení dalšího prohledávání.

Prohledávání do hloubky je opakem předchozího principu. V tomto případě se prohledávají jednotlivé větve až k finálnímu řešení v daném pořadí. Výhodou je brzké nalezení prvního řešení, nevýhodou je pak kvalita účelové funkce (obr 4-7).



Obr. 4-7: Částečný strom rozvrhovacího problému

Uvedený příklad stromu vychází z rozvrhovacího problému JSSP 3x3 uvedeného v kapitole 4.2 (tab. 4-1) a popisuje pouze jednu z možných větví. Větvení uvažuje omezení generování rozvrhů pomocí aktivního rozvrhu a nezobrazuje nereálná řešení či řešení nepodléhající druhu rozvrhu. V tabulce umístěné na obrázku je uvedena identifikace jednotlivých operací spolu s číslem udávajícím procesní čas (v závorce). Jednotlivé uzly jsou pak číslovány dle pořadí výskytu a obsahuje doplňkovou informaci číslo operace. V případě, kdy se nejedná o rozvrhovací uzel je doplněno pouze číslo operace.

Výhodou této metody je, že je schopna prohledat všechna reálná či nereálná řešení (dle nastavení fáze větvení) a nalézt tedy řešení optimální. Nevýhodou je vysoký výpočetní čas, který exponenciálně narůstá s lineárním nárůstem počtu operací resp. počtem rozhodovacích uzlů. Z tohoto důvodu se používá pouze v omezené míře. Nejčastější uplatnění tato metoda nalézá v případě teoretických problémů, které se používají pro testování úspěšnosti ostatních metod (resp. jejich algoritmů).

4.3.2 Matematické programování

V oblasti matematického programování v oblasti rozvrhování se používá několik metod, které se často v závislosti na řešeném problému kombinují [62].

Dynamické programování je přesná metoda, která prohledává celý prostor řešení odlišným způsobem než metoda větví a mezí. Tato metoda je založena na formulaci vratného procesu [40, 98], ve kterém je definováno, jak se k -tá iterace vypočte pomocí $(k-1)$ iterace a současně s tím musí být definován proces pro přímý výpočet iterace výchozí.

Tento proces je založen na dvou klíčových idejích:

1. princip optimality,
2. vratný princip optimality.

Princip optimality říká, že nejlepší cesta (zdroje na zdroj) z A do B k Z je nejlepší cestou z A do B a z B do Z. Vratný princip optimality pak říká, že nejlepší cesta z B do Z nemusí být známá a bude odhadnuta na základě použití principu optimality do té doby, než bude vypočtena exaktně [98]. Hlavním problémem je pak odlišnost jednotlivých problémů, tzn. nelze nastavit sestavení předpisů obecně. Pro každý problém je nutné vyvinout jedinečný předpis. To vše v závislosti na optimalizační funkci a omezení (technologický postup, seřizování, uvolňování materiálu do výroby atd.).

Další metodou z této oblasti je Lagrangeova relaxace [99], která se používá k rozložení problému na subproblémy. Pomocí iterativního řešení těchto sub problémů a aktualizací Lagrangeových multiplikátorů lze nalézt dolní meze daného řešení. Tato relaxace se v oblasti rozvrhování používá zejména pro problémy spojené s volbou pořadí operací a velikosti výrobní dávky.

Tato metoda, stejně jako ta předešlá, je velmi náročná na výpočetní čas a navíc, až na malé teoretické problémy nebyla stanovena optimální idea vratného principu optimality [99].

4.4 Přibližné metody řešení

V rámci řešení kombinatorických problémů, jako je rozvrhování, je obvyklé, že vzhledem k nedostatku času není možné dosáhnout optimálního řešení. Pro řešení se tak často využívají přibližné, heuristické metody. Jedná se o konstruktivní algoritmy fungující na základě pevně stanovených pravidel, dále metody založené na prohledávání okolí a to slepě se silnou stochastickou složkou či prohledávací strategií založené na znalosti o problému.

4.4.1 Konstruktivní algoritmy

Konstruktivní algoritmy se používají jako první z metod pro řešení rozvrhovacích problémů. Konstruktivní algoritmy jsou zpravidla jednoduché, zřetelné a tudíž se dají jednoduše vysvětlit i pracovníkům na dílně. Jsou nejčastěji používané pro řešení těchto problémů v praxi. CA začínají s prázdným řešením (plánem) a použitím metod, jako jsou prioritní pravidla [100], rozvrhování dle úzkého místa [15] či dle zakázky s největší sumou procesních časů [101]. Řeší rozvrhovací problémy respektující omezení (technologický postup, materiálové požadavky, požadavky na seřízení stroje) v konečném počtu kroků.

4.4.1.1 Rozvrhování pomocí prioritních pravidel

Rozvrhování podle prioritních pravidel je založeno na generování rozvrhů pomocí algoritmu Gifflera a Thompsona [102] (aktivní rozvrh) a jeho modifikace pro

rozvrh bez zpoždění. V těchto algoritmech se rozvrhují jedna operace po druhé na základě výběru právě dostupných operací. Právě dostupnou operací lze nazvat takovou, která je nerozvrhnutá a je jako první v technologickém postupu. V případech reálných rozvrhů je pak potřeba brát v potaz další omezení, jako je čas uvolnění materiálu do výroby, směnnost, připravenost pracovníka aj. Z takto dostupných operací se vybírá stroj, který obsahuje operaci, u které se předpokládá nejdřívější čas dokončení (či začátku pro rozvrh bez zpoždění). Na tomto stroji je pak nutné rozhodnout, jaké z dostupných operací dát přednost.

Toto rozhodnutí lze stanovit na základě prioritních pravidel vyplývajících z:

- obecných požadavků na výrobu,
- vlastností daného rozvrhu – technologického postupu.

V praktických aplikacích se často vyskytují prioritní pravidla dle požadavků na výrobu. Informační systémy ERP/APS často umožňují ruční přiřazování priorit dané zakázce, kde se stanovuje priorita na základě těchto informací:

- požadovaný termín odvedení zakázky zákazníkovi,
- termín přijetí zakázky,
- penále z nedodržení termínů,
- druh zakázky (zhromadněné či nezhromadněné díly).

V případě zhromadněných dílů pak plánovací systémy dynamicky zvyšují prioritu dle vzdálenosti k objednáci či kritické hladině a předpokládané spotřebě.

Přes to, že takový to způsob nejčastěji odpovídá požadavkům podniku, není příliš vhodný pro dílenské řízení výroby. V případě, že je nutné vyrábět na jednom stroji dvě zakázky, které mají např. stejný datum dodání zákazníkovi a tudíž stejnou prioritu, priorita neurčí pořadí zpracování na takovém stroji.

V těchto případech se stále častěji používají prioritní pravidla vycházející z vlastností daného rozvrhu resp. technologického postupu. Výzkum v této oblasti je rozsáhlý a již v 70 letech minulého století jich bylo známo více než sto [100].

Mezi nejznámější pravidla patří:

- SPT (Shorter Processing Time) – zakázka s nejkratším procesním časem
- MWKR (Most Work Remaining) – zakázka, na které zbývá nejvíce práce
- LWKR (Least Work Remaining) - zakázka, na které zbývá nejméně práce
- FCFS (First Come First Served) známější pod názvem First-In First-Out (FIFO) neboli první dostupný je prvním zpracovaným – zakázka s nejdřívějším časem příchodu na pracoviště
- RS – (Random Select) - náhodný výběr zakázky

Výhodou tohoto způsobu rozvrhování je rychlé a jednoduché generování rozvrhů. Nevýhodou pak skutečnost, že nelze zaručit nalezení optimálního řešení a možnost, že v případě, kdy se vygenerují rozvrhy pomocí různých prioritních pravidel, lze dosáhnout stejného řešení. Nejvhodnější nasazení této metody je v případě, kdy je vyžadováno řešení v krátkém čase, pro výpočet rozvrhů v případě generování sekvencí

pomocí heuristických metod, nebo pro generování počátečních řešení pro prohledávací heuristické metody.

4.4.1.2 Heuristika hrdla lahve

Hrdlo lahve je termín, který označuje nejužší místo v systému resp. stroj, který svou produkcí definuje maximální čas dokončení všech zakázek. Hlavním principem této metody je přerozvrhování výrobních zakázek právě na úzkém místě, které se po vyřešení problému přesouvá na jiný výrobní zdroj. Tak jako konstruktivní algoritmus Gifflera a Thompsona je tento algoritmus iterativní s tím rozdílem, že nerozvrhuje zakázky jednu po druhé, ale tvoří fixovaný rozvrh pro samostatné stroje a to následujícím způsobem [15, 16]:

- Inicie - nalezení předpokládaného úzkého místa součtem všech procesních časů zakázek na jednotlivých strojích. Výběr sekvence zakázek na tomto zdroji a rozvrhnutí ostatních strojů odpovídajícím způsobem včetně nalezení maximální doby dokončení všech zakázek C_{max} .
- Nalezení kritické cesty a maximálního zpoždění - postupně pro každý stroj je vyřešen speciální rozvrhovací problém pro jeden stroj (berou se v úvahu zpoždění vlivem rozvržení zakázek na ostatních strojích) s termíny dostupnosti r_j , dokončení d_j a kritériem minimalizace maximálního zpoždění L_{max} .
- Výběr stroje s největším maximálním zpožděním L_{max} , nalezení jeho optimálního rozvrhu (dle zpoždění) a podřízení rozvrhů ostatních strojů. Tento krok obsahuje přepočty zpoždění L_{max} a maximálního data dokončení všech zakázek C_{max} . Tento krok lze opakovat, dokud nejsou všechny stroje rozvrženy, resp. na všech strojích je nulové zpoždění vzhledem k termínu dokončení.

Nevýhodou tohoto principu je, že se snaží o minimalizaci zpoždění vytvořeného počátečního řešení s ohledem na celkový čas dokončení. Ačkoli je tento způsob považován za lepší způsob generování rozvrhů než je tomu u prioritních pravidel [16], nezaručuje, že nalezne optimální řešení. Často je tento princip používán v kombinaci s genetickými algoritmy [18, 103].

4.4.1.3 Vkládací algoritmus

Vkládací algoritmy svým principem patří v oblasti teoretických problémů spíše mezi algoritmy prohledávací. V rozvrhování praktických úloh spíše mezi algoritmy konstruktivní. Hlavní rozdíl při pohledu na vkládací algoritmus vychází ze skutečnosti, že v praxi se spíše řeší příchod nových zakázek, než konstrukce rozvrhu od jeho počátku.

Vkládací algoritmy vycházejí z předpokladu, že již existuje částečný rozvrh a úlohou je vložit nově příchozí aktivity do rozvrhu.

Při řešení teoretických problémů se pak přistupuje ke generování částečných rozvrhů pomocí prioritních pravidel a to nejčastěji pomocí RS. Při optimalizaci takovýchto rozvrhů se vychází z fixovaných operací (již rozvržené) a operací, které je nutné rozvrhnout [104]. Operace se pak dále rozvrhují na základě velikosti procesních časů.

Dalším možným přístupem je vygenerování rozvrhu pomocí metody větví a mezí do náhodně definované hloubky a šířky. Následně se prochází otevřené

rozhodovací uzly (uzly, kde je možné dále vkládat operace) a vkládají se postupně operace podle nejmenšího zvětšení celkové doby odvedení všech zakázek C_{max} [105].

Parametrem, řídícím vkládací algoritmus je tedy poměr rozvržených a nerozvržených operací, který jednak definuje šířku (hloubku) prohledávání, výpočetní čas a dosažené optimum. V případě rozvrhovacích algoritmů je dosahováno nejlepších výsledků s poměrem 10/1 (rozvržené/ vkládané).

Na tomto principu se vyvíjí prohledávací metody, kde po konstrukci rozvrhu se vyjímá daný počet operací z kritické cesty a opět vkládá v rozvrhovacích uzlech dle nejmenšího nárůstu účelové funkce.

4.4.2 Metody založené na lokálním prohledávání

Metoda lokálního prohledávání patří mezi nejstarší metody založené na prohledávání oblasti řešení [106]. Patří mezi nejjednodušší a nejuniverzálnější metody. Pomocí definované oblasti řešení, účelové funkce a mechanismu generování sousedních řešení se snaží dosáhnout globálního extrému.

Postup optimalizace je pak následující:

- 1) Vygenerování počátečního řešení pomocí konstruktivního algoritmu (nejčastěji pomocí pravidla RS).
- 2) Vygenerování sousedních řešení a výpočet účelové funkce.
- 3) Výběr nejlepšího řešení dle účelové funkce.
- 4) Stanovení tohoto řešení jako počátečního a návrat na krok č. 2, dokud není splněna podmínka ukončení algoritmu.

Ukončení tohoto algoritmu obvykle závisí na počtu nadefinovaných iterací či na ukončující podmínce uvíznutí v lokálním extrému.

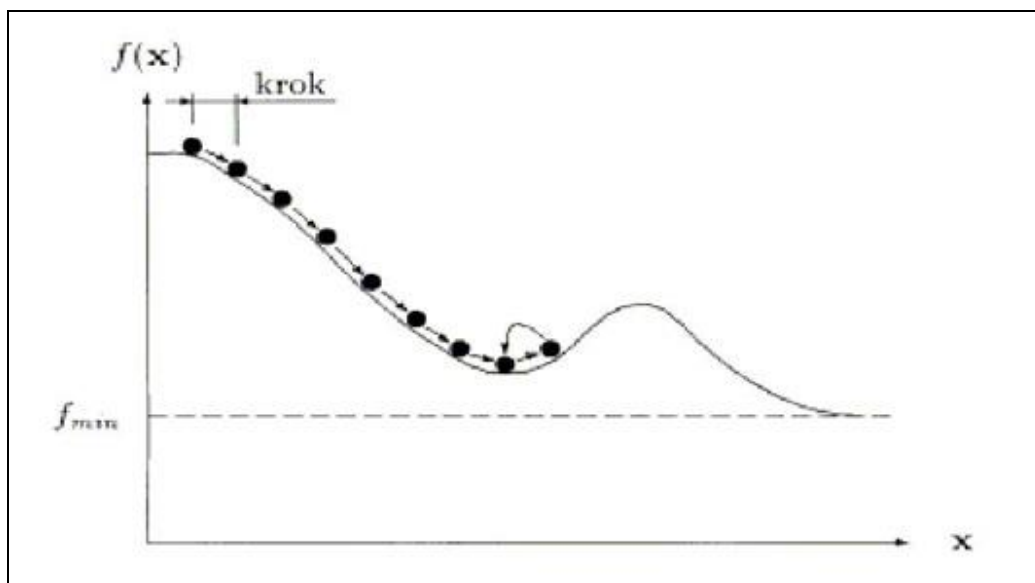
Nevýhodou této metody je časté uvíznutí v lokálním extrému, což je způsobeno počátečním řešením. Tento problém se obvykle řeší pomocí:

- Zvětšení množiny sousedů.
- Opakování metody pro různá náhodně vygenerovaná počáteční řešení.
- Přípuštění přijetí řešení s horší hodnotou účelové funkce.

Přes to, že se jedná o metodu často končící v lokálním extrému, mnohé principy, zvláště v oblasti generování sousedství, se využívají v pokročilých optimalizačních metodách [25, 29].

4.4.2.1 Horolezecký algoritmus

Horolezecký algoritmus podobně jako lokální prohledávání pro vygenerované řešení generuje okolí (množinu sousedních řešení), hledá optimální hodnotu účelové funkce v tomto sousedství a zaznamenává nejlepší řešení. Hlavní rozdíl oproti lokálnímu prohledávání je, že horolezecký algoritmus není ukončen při nalezení globálního extrému. V tomto případě je nadefinovaný určitý počet iterací do cílového prostoru resp. řešení (lze připodobnit k táboru při lezení na horu), které je použito jako střed nově prohledávaného prostoru. V tomto okolí se hledání optimálního řešení opakuje a to i v případě, že dané řešení (tábor) má horší hodnotu účelové funkce než je aktuálně nejlepší řešení (obr. 4-8).



Obr. 4-8: Příklad horolezeckého algoritmu na jednoduché funkci [107]

Nevýhodou horolezeckého algoritmu je možnost jeho zacyklení. Je totiž možné, že se algoritmus navrátí do řešení, které bylo nastaveno jako výchozí v předešlých iteračních krocích. Problém zacyklení se řeší jako u lokálního prohledávání např. opakováním metody pro různá náhodně vygenerovaná počáteční řešení.

Další z možností jak eliminovat zacyklení a dosáhnout lepších hodnot účelové funkce je použít pokročilejší formy tohoto algoritmu jako je:

- Horolezecký algoritmus s učení [107] – obsahuje pravděpodobnostní parametr, který určuje rychlost zlepšování účelové funkce a zastaví algoritmus v případě, že se účelová funkce nezlepšuje dostatečně rychle.
- Stochastický horolezecký algoritmus [108] – na místo volby řešení s nejlepší hodnotou účelové funkce jako nového výchozí řešení volí náhodně.
- Paralelní horolezecký algoritmus [109] – prohledávání ne z jednoho, ale z více výchozích řešení.

Ačkoli se jedná o pokročilejší metodu založenou na lokálním prohledávání, spíše než samostatně se využívá v oblasti hybridizace pokročilejších algoritmů, jako jsou například genetické algoritmy [110].

4.4.2.2 Metoda simulovaného žíhání

Metoda simulovaného žíhání, stejně jako horolezecký algoritmus, je metodou založenou na lokálním prohledávání a připouští přijetí takového řešení, které má horší hodnotu účelové funkce [40, 111].

Krystal se zahřeje na určitou (vysokou) teplotu a potom se pomalu ochlazuje (žíhá). Defekty krystalické mřížky mají při vysoké teplotě vysokou pravděpodobnost zániku. Pomalé ochlazování systému zabezpečí, že pravděpodobnost vzniku nových defektů klesá. Při žíhání se soustava snaží dostat do takového stavu, ve kterém je její energie minimální – tj. krystal bez defektů. Terminologie této metody vychází z předcházející analogie [40, 112]:

- Žihání – kombinatorická optimalizace
- Stav systému - přípustné řešení (x)
- Energie stavu systému – hodnota účelové funkce ($f(x)$)
- Změna stavu systému – přechod k sousednímu řešení (x^*)
- Teplota – řídicí parametr (T)
- Ustálený stav systému – přijaté řešení

Postup optimalizace pomocí této metody je pak následující:

Kde T_0 je počáteční teplota, T_f teplota konečná a $T(t)$ je funkce poklesu teploty, n_T pak počet opakování algoritmu pro danou teplotu T .

Vygenerování počátečního řešení x_0 (nejčastěji pomocí konstruktivního algoritmu a prioritního pravidla RS)

```

 $x^* = x_0$ 
set  $T_0 > 0$ 
 $T = T_0$ 
set  $a(t)$  and  $T_f$ 
While  $T < T_f$ 
  for  $i = 1$  to  $n_T$ 
    náhodně vyber  $x$  z množiny sousedních řešení  $S(x_0)$ .
     $\Delta f = f(x) - f(x_0)$ 
    if  $\Delta f < 0$  than
       $x_0 = x$ 
      if  $f(x) < f(x^*)$  than
         $x^* = x$ 
      endif
    else
      náhodně vyber  $r$  z rovnoměrného rozdělení na intervalu (0,1)
      if  $r < e^{-\Delta f/T}$  than
         $x_0 = x$ 
      end if
    end if
  next i
   $T = T(t)$ 
end while

```

Jak je zřejmé, při optimalizaci tímto algoritmem je vždy přijato takové řešení, které má lepší hodnotu účelové funkce než to stávající. Řešení horší než stávající má pravděpodobnost přijetí danou Metropolisovým kritériem P .

$$P = e^{-\frac{f(x) - f(x_0)}{T}} \quad (12)$$

Rychlost výpočtu a kvalita nalezených řešení je pak závislá na následujících parametrech [40, 111, 112]:

- Parametry vázané na optimalizovaný problém (složitost problému, způsob generování sousedních řešení, účelová funkce).

- Parametry optimalizační metody simulovaného žihání (počáteční T_0 a konečná T_f teplota, funkce redukce teploty $T(t)$ a počet opakování pro danou teplotu n_T).

Metoda simulovaného žihání má větší pravděpodobnost nalezení globálního extrému než horolezecký algoritmus vzhledem k možnosti uniknout z lokálního extrému na větší vzdálenost než o jedno řešení a nedochází tedy u této metody k tak častému zacyklení. Tato metoda resp. její principy se používají spíše u pokročilejších algoritmů jako je zakázané prohledávání a u genetických algoritmů [113].

4.4.2.3 Metoda zakázaného prohledávání

Tato metoda vychází ze dvou základních principů a to principu lokálního prohledávání a umělé inteligence, kterou aplikuje pomocí paměti uchováající předešlá řešení. V této paměti se uchovává několik posledních transformací (v případě rozvrhování změn pořadí operací na zdrojích), pomocí kterých byla generována sousední řešení toho stávajícího. Tímto mechanismem se omezuje problém zacyklení z metody horolezeckého algoritmu. Poslední transformace se uchovávají ve formě seznamu řízeného prioritním pravidlem FIFO a má omezenou délku. V případě zaplnění tohoto seznamu tedy nově přichozí způsobuje smazání transformace, která je v seznamu nejdéle [40, 114, 115].

Tento seznam slouží jako omezení (tabu) prohledávaného sousedství (zakazuje určité cesty v uzlových bodech rozhodování. Díky transformacím, které jsou pro následující prohledávání tabu, se nejčastěji tato metoda nazývá tabu prohledávání (z angl. Tabu Search). Omezení sousedství lze prolomit pouze aspiračním kritériem (pravděpodobností, že bude přijato horší řešení), což je případ, kdy zakázaná transformace by v budoucnu zlepšila hodnotu stávajícího nejlepšího řešení.

Nejdůležitější parametr tohoto algoritmu je pak velikost „tabu seznamu“. Velikost „tabu seznamu“ musí být kratší, než je počet možných transformací v rámci sousedství. Při malé velikosti seznamu dochází k zacyklení, podobně jako tomu je u horolezeckého algoritmu, avšak s tím rozdílem, že ne v navazujících dvou transformacích (minimální délka tabu seznamu = 1). Se zvětšováním velikosti tabu seznamu se pak zvětšuje pravděpodobnost, že se přeskočí takové transformace, které mohou vést k získání globálního optima.

Vzhledem k tomu se často zavádí parametry modifikující pravděpodobnost zakázání transformací. Jedním z možných parametrů je pak proměnná velikost tabu seznamu měnící se podle průběžných výsledků. Další možností je využití seznamu frekvencí daných transformací (v podobě inverzních transformací nejlepších řešení), kde se upravuje „zakázání“ transformací úměrně s frekvencí jejich použití (tabu list). V tabu seznamu se cyklicky obměňují použité transformace a v tabu listu se penalizují řešení s největší frekvencí v tabu seznamu.

V oblasti rozvrhování dílenských zakázek ovlivňuje zásadně kvalitu řešení také mechanismus transformací. Mezi nejznámější transformace patří Bruckerova transformace [96] a transformace Nowicki a Smutnicki [116]. Obě vychází z prohledávání sousedních řešení změnou pořadí zakázek na kritické cestě.

Kombinace zakázaného prohledávání a transformace Nowicki a Smutnicki patří dodnes k nejúspěšnějším algoritmům v oblasti řešení teoretických problémů typu JSSP a patří mezi nepoužívanější hybridizace evolučních algoritmů.

4.4.3 Metody založené na evolučních principech

Evoluční algoritmy (EA) jsou meta-heuristické optimalizační algoritmy založené na optimalizaci populace (množiny řešení) pomocí principů známých z biologie jako jsou mutace, křížení, přirozený výběr a přežití nejsilnějších jedinců za účelem nalezení nejlepších jedinců skrze generace (iterace).

V roce 1859 Charles Darwin publikoval práci „O původu druhů“, ve které pojmenoval objevené principy přirozené selekce a přežití nejsilnějších jedinců jako hybné síly evoluce. Jeho dedukce lze shrnout do následujících bodů:

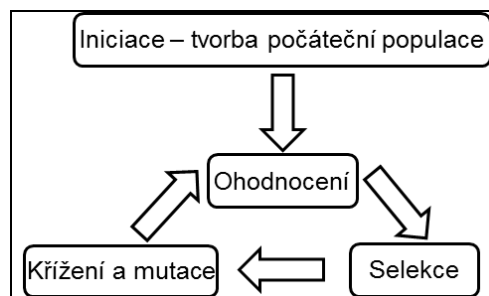
- Jedinci v rámci jedné populace jsou velmi plodní a plodí více potomků, než je schopno dospět.
- Pokud do populace nezasahují vnější vlivy (přírodní katastrofy, lidský vliv atd.), velikost populace bývá neměnná.
- Pokud nezasahují vnější vlivy, zdroje obživy (jídlo) bývají stabilní v čase.
- V případě, že někteří jedinci soupeří o tyto limitované zdroje, populace se začne strachovat o přežití.
- Zvláště v oblasti sexuální reprodukce, žádný z jedinců si není rovný (podřízenost a nadřízenost jedinců jak sociální, tak fyzická).
- Rozdílná síla reprodukcí se jedinců způsobuje variantnost potomků jejich síly a schopnosti přežít.
- Tyto schopnosti jsou dědičné.
- Slabší jedinci mají menší pravděpodobnost než ti silní, že se budou reprodukovat a zplodí silné potomky.
- Zvláštnosti, které nemají na první pohled vliv na přežití, se přenášejí z rodičů na děti.
- Druh se pomalu mění a přizpůsobuje více a více prostředí, ve kterém žije a v rámci tohoto procesu se může nakonec vyvinout nový druh.

Evoluční algoritmy vycházejí z těchto biologických principů a principu cílené optimalizace tj. zlepšování určitého znaku jedinců – účelové funkce. Prohledávaný prostor evolučních algoritmů G je abstrakce množiny všech možných DNA řetězců v přírodě a jejich znaky $g \in G$ (v rozvrhování pořadí operace) hrají roli přírodního genotypu. Populace jedinců x (řešení problému) se vyvíjí (prohledává prostor X) pomocí mutace a křížení předem definovaným směrem daným nutnou silou pro přežití (účelová funkce).

4.4.3.1 Genetické algoritmy

Genetické algoritmy jsou podmnožinou evolučních algoritmů, kde jedinci (řešení) v prohledávaném prostoru G jsou reprezentovány binárním řetězcem ($G=B^*$) resp. vektorem. Genetické algoritmy (GA) jsou adaptivní heuristické algoritmy popsané J. H. Hollandem [117]. Výraz genetické algoritmy bývá často nesprávně používán pro celou třídu evolučních algoritmů a to zejména ve spojitosti s algoritmy cílenými na konkrétní problémy (např. rozvrhování) tj. evoluční strategie.

Genetické algoritmy, tak jako většina evolučních algoritmů, mají následující řadu kroků (obr. 4-9).



Obr. 4-9: Postup při optimalizaci GA

V rámci těchto kroků:

- Inicializace – při tvorbě počátečních řešení se náhodně generuje řada genů - chromosom, který v tomto případě má binární formu (tab. 4-2).

Tab. 4-2: Chromozom jednoho jedince

1	0	0	1	1	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---

- Ohodnocení – Výpočet řešení jedné či více účelových funkcí f a přidělení hodnoty vhodnosti F pro křížení pro všechny jedince zvlášť (tab. 4-3).

Tab. 4-3: Ohodnocení populace

Jedinec	Gen										Síla		
	1	2	3	4	5	6	7	8	9	10	f_1	f_2	F
1	1	0	0	1	1	0	0	1	0	1	10	2	12
2	0	1	0	0	1	0	1	1	0	0	12	8	20
3	0	0	1	1	0	1	1	0	1	1	5	11	16
4	1	1	0	0	0	1	0	0	1	1	7	8	15

- Selektce – V selekci probíhá eliminace jedinců, kteří nejsou schopni pokračovat do další generace (princip elitismu – do další reprodukce generace vstupuje nejsilnější jedinec jako první) a výběr jedinců (rodičů) pro křížení. Existuje několik postupů výběru rodičů jako např. ruletová (tab. 4-4), turnajová nebo pořadová selektce více viz kapitola 4.4.3.3.

Tab. 4-4: Pravděpodobnost výběru jedince pro reprodukci ruletovým pravidlem

Jedinec	Gen										Síla			P. Výběru %
	1	2	3	4	5	6	7	8	9	10	f_1	f_2	F	
1	1	0	0	1	1	0	0	1	0	1	10	2	12	80,95%
4	1	1	0	0	0	1	0	0	1	1	7	8	15	57,14%
3	0	0	1	1	0	1	1	0	1	1	5	11	16	31,75%
2	0	1	0	0	1	0	1	1	0	0	12	8	20	0,00%

- Křížení a mutace – Křížení znamená prohození částí chromosomů (s danou pravděpodobností) rodičů jednobodově (obr. 4-10), dvoubodově či vícebodově – uniformně. V operaci mutace se změní na základě dané pravděpodobnosti jeden či více genů. Tak jako u křížení existuje jednobodová, dvoubodová a vícebodová mutace. U GA mutace znamená změnu hodnoty genu z 1 na 0 či naopak.

Jedinci	Chromosomy rodičovského páru										
1	1	0	0	1	1	0	0	1	0	1	
4	1	1	0	0	0	1	0	0	1	1	
				Bod křížení							
				Chromosomy potomků							
				1	1	0	1	1	0	0	1
				1	0	0	0	0	1	0	1

Obr. 4-10: Jednobodové křížení

Operátory ovlivňující optimalizaci jsou pak:

- Velikost populace (počet jedinců – řešení v jedné generaci)
- Počet generací (počet iterací algoritmu)
- Princip selekce, křížení a mutace
- Pravděpodobnost křížení a mutace

Genetický algoritmus se stal velmi populárním nástrojem optimalizace pro svou schopnost překonávat lokální extrémy. Tato vlastnost je však zásadně ovlivněna jejím vhodným nastavením řídicích operátorů. Pro řadu optimalizačních problémů se GA využívají dodnes, avšak vývoj a aplikace EA vedla k specializaci algoritmů směrem k specifickým problémům a tedy k pokročilejším metodám jako jsou tzv. evoluční strategie.

4.4.3.2 Genetické programování

Genetické programování stejně, jako GA, vychází z principů evoluce. Na rozdíl od GA genetické programování ve svém chromosomu nereprezentuje daný optimalizační problém, nýbrž metody určené pro zpracování daného subproblému. Jedná se tedy o metodu strojového učení, která používá evoluční algoritmy, které postupně zlepšují populaci počítačových programů (podprogramů – procedur) [40].

Základní pojmy:

- Jedinec - vhodně zakódovaný počítačový program. Během výpočtů se takových programů vygeneruje v řadě po sobě následujících generací velké množství a na závěr se vybere nejlepší z nich.
- Zdatnost čili fitness - výsledek funkce zdatnosti (účelová funkce či ohodnocení více funkcí), která každému jedinci přiřadí číslo vyjadřující jeho schopnost řešit původně zadanou úlohu. Čím vyšší zdatnost, tím je jedinec kvalitnější. Hledaný nejlepší jedinec se pozná právě podle toho, že dosáhl nejvyšší zdatnosti. Funkce zdatnosti tak vyjadřuje cíl celého počínání, tj. program kvalitně řešící zadanou úlohu.
- Populace je soubor jedinců, které počítač zpracovává v jedné iteraci (generaci).

Stejně jako jiné kombinatorické problémy má prohledávaný prostor S stromovou strukturu a své omezení [118, 119]. To lze demonstrovat např. na řešení otázek, jaké operátory v rámci evolučního algoritmu použít. V předchozích kapitolách byly zmíněny základní operátory princip selekce, křížení a mutace, které ovlivňují výkonost EA. V tomto případě lze tedy zavést chromosom obsahující čtyři geny, které mohou nabývat různých hodnot (tab. 4-5).

Tab. 4-5: Genotyp genetického algoritmu

Pořadí chromosomu	Funkce subprogramu	Vlastnost subprogramu	Reprezentace subprogramu
1	Selekce rodičů	Ruletové kolo	a
		Pořadová selekce	b
2	Selekce jedinců do další generace	Turnajová selekce	a
		Nejlepší řešení a noví potomci	b
		Lepší polovina nové a staré generace	c
3	Křížení	Jedno bodové	a
		Dvou bodové	b
		Uniformní	c
4	Mutace	Jedno bodová	a
		Dvou bodová	b
		Více bodová	c

Na rozdíl od GA u GP se nekóduje problém do řetězců pevné délky, ale do struktur s variabilní délkou. Tzn. rozhodovací strom a popis problému může být jako na obrázku 4-7 (kapitola 4.3.1). Je nutné si uvědomit, že jednotlivé členy (a,b,c atd.) mají variabilní délku.

Populace takového problému zobrazuje tabulka 4-6.

Tab. 4-6: Populace v rámci genetického programování

Jedinec	Gen			
	1	2	3	4
1	a	a	b	b
2	b	b	a	c
3	a	c	c	a
4	b	a	b	a

Genetické programování je metodou, která umožňuje najít řešení klasických problémů a někdy také najde uspokojivé řešení kombinatorických úloh (rozvrhování). Nejčastější aplikací GP je v oblasti data-miningu či tvorby předpovědí na základě historických dat. GP může být u některých typů úloh (jako je rozvrhování či problém obchodního cestujícího) náročnější na výpočetní výkon ve srovnání se specializovanějšími metodami a jejich algoritmy.

4.4.3.3 Evoluční strategie

Evoluční strategie ES byly poprvé uvedeny Rechenbergem a jsou to speciální optimalizační metody založené na idejích adaptace a evoluce tak jako GA. Od GA se však liší v těchto bodech [120]:

- Reprezentace problému není kódovaná pomocí binární soustavy, ale pomocí reálných čísel reprezentujících dané problémy (u rozvrhování např. číslem zakázky či operace).
- Mutace a selekce je v těchto metodách zásadní, křížení v nich nemá tak důležitou roli.
- Mutace genu $x(i)$ daného řešení x je prováděna většinou náhodně pomocí normálního rozdělení $N(x(i), \sigma^2_i)$.

Dále evoluční strategie obvykle kombinují přechod více jedinců (ne pouze toho nejlepšího) ze staré generace do generace nové spolu s následujícími přístupy k selekci [121,122].

(1+1) – ES

Populace obsahuje pouze jednoho jedince. V následném souboji (rodič – potomek) postupuje lepší z nich do další generace. Tato strategie je velmi podobná lokálnímu prohledávání a horolezeckému algoritmu.

(μ +1) – ES

V této strategii populace obsahuje μ jedinců, ze kterých je náhodně vybrán jeden. Tento jedinec se vybírá z populace tvořené stávající populací a populací potomků. Nejhorší jedinec se z populace odstraňuje.

(μ + λ) – ES [123]

Pomocí reprodukční operace z vybraných μ rodičovských řešení vznikne $\lambda < \mu$ potomků. Ze společného seznamu $\mu+\lambda$ řešení se vybere pouze μ nejlepších jedinců, kteří postoupí do další generace.

(μ, λ) – ES [124]

Stejně jako v předchozí strategii je vygenerováno $\lambda < \mu$ potomků z μ rodičů, avšak rodiče jsou postupně odstraňováni nejlepšími jedinci z λ tak, aby postoupilo do další generace μ jedinců.

$(\mu/\rho, \lambda)$ – ES

Tato strategie je v podstatě stejná jako (μ, λ) – ES s přidáním parametru ρ , který udává počet rodičů jednoho potomka. Potomek tedy vzniká rekombinací několika rodičů. V této oblasti figuruje několik obměn jako např. $(\mu/\mu, \lambda)$ – ES či $(\mu/\rho + \lambda)$ – ES.

$(\mu', \lambda'(\mu, \lambda)^\gamma)$ – ES [125]

Jedná se o tzv. vnořenou strategii, kde je vytvořeno λ' potomků z μ' , kteří jsou izolovány na γ generací. V každé γ generaci je vytvořeno λ potomků, ze kterých je vybráno μ jedinců do další generace. Po γ generaci je vybrán nejlepší potomek z každé izolované populace a je zařazen zpět do reprodukční populace. Poté se celý cyklus opakuje s λ' potomky.

Postupy v evolučních strategiích se neustále rozvíjí a patří mezi nejperspektivnější přístupy v optimalizaci. Nevýhodou dalšího vývoje je jeho obtížná sledovatelnost z důvodu nesprávného zaměňování pojmu evoluční strategie za genetický algoritmus.

4.4.3.4 Memetické a hybridní algoritmy

Hybridizace evolučních algoritmů jinými optimalizačními metodami jako jsou lokální prohledávání, simulované žíhání či horolezeckým algoritmem mají dlouhou tradici [126,127,128].

Memetické algoritmy stejně jako GA se snaží simulovat přirozenou evoluci s cílem vyřešit optimalizační problém. Termín memetické algoritmy byl definován poprvé Moskatem [129] jako přirovnání k simulacím sociální evoluce, kde vzorce chování přecházejí z generace na generaci pomocí tzv. memů (z řeckého mimema – napodobovat) - kulturních ekvivalentů genů.

Memetické (hybridní) algoritmy se opírají o následující principy:

- Inteligentní optimalizace v rámci jiných algoritmů pomocí metod založených na lokálním prohledávání.
- Soutěžení daných řešení na základě principů selekce (tak, jak je tomu u evolučních strategií).
- Kooperace ve formě křížení se znalostí specifických vlastností optimalizačních problémů.

Definice memetických algoritmů daná Moskatem [129] je relativně obecná a shoduje se s jinými algoritmy (často nelze odlišit např. memetické algoritmy a evoluční strategie). V oblasti hybridizace pak existují dva základní směry a to Lamarckizmus a Baldwinův efekt.

Lamarckianův princip [130] evoluce je znám již před objevením genetiky a vychází z dvou základních principů:

- Jedinci mohou získat nové zvýhodňující vlastnosti (vzorové chování) a ztratit ty, co nepoužívají.
- Tyto vlastnosti přenášejí na své potomky. Tento bod byl pak vyvrácen výzkumem v oblasti genetiky.

Lamarckianův evoluční princip lze aplikovat v evolučních algoritmech např. pomocí aplikace lokálního prohledávání na každého nového jedince v rámci operace křížení či mutace.

Baldwinův efekt [131] je založen na spolupráci principů evoluce a učení. Tyto dva principy soutěží na základě ceny a výhody získané učením:

- Celoživotní učení dává jedinci šanci na adaptaci na životní prostředí i za možné změny fenotypu (spolupůsobení genotypu a prostředí) tj. získání nových znalostí a dovedností. Na rozdíl od Lamarckianovy evoluce tyto vlastnosti neovlivňují genotyp a nejsou dědičné.
- Evoluce postupně vytváří jedince, kteří jsou schopni se rychleji a jednodušeji učit tyto znalosti a dovednosti, až k dospění do stádia, kdy se tyto dovednosti a znalosti zakódují do jejich genotypu. Informace v genotypu pak nahradí učení. Tento proces se nazývá genetickou asimilací.

V praxi pak tyto změny neovlivňují genotyp, ale omezují prohledávaný prostor [132].

Prvky memetických algoritmů patří spolu s evolučními strategiemi mezi zásadní směry při optimalizaci pomocí evolučních algoritmů. V memetických algoritmech poskytují evoluční principy prohledávání prostoru do šířky, principy lokálního prohledávání pak umožňují jeho dostatečnou hloubku tj. nalézání lokálních a globálních extrémů.

4.4.4 Metody založené na umělé inteligenci

Obor umělé inteligence se snaží konstruovat inteligentní stroje a zprostředkovaně simulovat inteligentní chování živých organismů. Existuje několik tradičních pohledů na umělou inteligenci a to logicko-symbolická, symbolicko-reprezentační, algoritmická nebo komputacionalistická inteligence. Tradiční umělá inteligence zaujímá pohled na inteligenci shora, tedy od lidských intelektuálních schopností. Oproti tomu nový pohled vychází z chování jednoduchých organismů.

Nový přístup k umělé inteligenci se dělí na následující principy:

- Emergentní funkcionalita - racionální fungování agentů (systém, který vnořený do svého prostředí dosahuje na základě vjemů ze svých senzorů a činnosti svých aktuátorů určité cíle) může vzniknout samovolně v rámci intenzivních interakcí s prostředím.
- Dekompozice na úrovni úloh, kde je systém rozložen na komponenty, které jsou zhotoveny jen pro určitý úkol, za jehož splnění zodpovídají.
- Reaktivita, která je bezprostřední reakcí na aktuální stavy prostředí.

Výzkum v oblasti umělé inteligence a optimalizace kombinatorických problémů se principiálně opírá o výzkum v oblasti neuronových sítí a evolučních algoritmů.

Na neuronové síti lze pohlížet jako na jednoduše komunikující elementy, jejichž způsob komunikace je odvozen z analogie ke komunikaci neuronů v centrální nervové soustavě.

V oblasti optimalizace problémů se stromovou strukturou lze hovořit o zpracovatelských jednotkách, jejichž spojení s ostatními je ohodnoceno svojí váhou. Jednotky přijímají přes tyto spoje numerické informace. Na základě informace tyto jednotky spouští pomocí aktivační funkce svoji činnost. Typ aktivační funkce pak rozhoduje o činnosti jednotky. Jednotky mají pak základní dvě funkce a to pamatování informací a činnost závisící na informaci. Přes to, že neuronové síti skrývají velký potenciál, jejich nevýhodou je neexistence rigorózní metody pro aplikaci na jednotlivé problémy.

Prvky umělé inteligence jsou pak aplikovány s ohledem na pozorování chování organizovaných systémů v přírodě, jako tomu je u imunitních algoritmů, algoritmů rojení částic či chování mravenců v koloniích.

4.4.4.1 Imunitní algoritmy

Umělé imunitní systémy (AIS) mohou být definovány jako distribuční a adaptivní výpočetní systémy inspirované lidským imunitním systémem a mohou být aplikované na specifické problémy. Oblast výzkumu imunitních algoritmů je poměrně nová a v současné době je považována za jeden z trendů výpočetních nástrojů v různých oblastech vědy [133, 134].

Hlavní prvky imunitních systémů a jejich algoritmů je [135]:

- Učení - znak dospělosti, který zaručuje rozpoznávání vzorců chování modelů optimalizačních problémů.
- Adaptace - buňky v imunitním systému jsou tvořeny na základě souboje s poškozujícími antigeny nahrazováním starších, neefektivních či mrtvých buněk tak, aby organismus přežil.
- Distribuce - každá buňka imunitního systému má vlastní distribuční funkci, která poznává a reaguje na cizí antigeny, které mohou napadnout buňku v jakémkoli místě.
- Samo-organizující - imunitní systém má vlastní samo-organizující mechanismus, který napomáhá kontrolovat populaci lokálními interakcemi tak, aby udržel rovnovážný stav v celém organismu.
- Rozmanitost - existují dva hlavní procesy udržení rozmanitosti. První proces obsahuje produkci skoro nekonečných typů receptorových molekul pomocí rekombinace konečné množiny genů. Druhý proces pomáhá reprodukci imunitních receptorů (somatická hypermutace).
- Rozpoznávání - imunitní systém organizuje, identifikuje a reaguje na jiné buňky. Tzn. má schopnost rozpoznat vlastní a cizí buňky.
- Paměť - organismus na základě rozpoznání dovolí některým buňkám a molekulám žít déle, aby byly dostupné k boji proti budoucím infekcím (v biologii zastává tento princip vakcinace).

Postup imunitního algoritmu je pak následující:

- 1) Tvorba protilátek (populace).

- 2) Dekódování protilátek a výpočet afinity (účelové funkce).
- 3) Výpočet pravděpodobnosti generování protilátek (princip selekce – klonování).
- 4) Mutace (v případě rozvrhování prohazování operací) pro vznik nových klonů.
- 5) Výběr nejlepších klonů tak, aby nahradily ty nejhorší.
- 6) Opakování procesu dokud není dosaženo požadovaného množství generací.

V oblasti rozvrhování umožňuje výpočet afinity (13) rozdílnost populace. V případě, že je účelovou funkcí makespan, pak (kde y je rozvrh -protilátka):

$$Afinita(y) = \frac{1}{makespan} \quad (13)$$

V porovnání s genetickými algoritmy (tab. 4-7) je pak největším rozdílem mechanismus tvorby nových potomků (klonů), mutace na základě afinity a paměť (přežívání na základě definovaného stáří).

Tab. 4-7: Porovnání GA a AIS v analogii k rozvrhování [136]

GA	AIS	Analogie s rozvrhováním
Chromosom: komponenty reprezentující systém	Protilátky	Rozvrh
Gen: díl chromosomu	Molekuly receptorů	Zakázka (či operace) v rozvrhu
Účelová funkce: ohodnocení každého jedince	Afinitní funkce	Odpovídající kritérium rozvrhu např. makespan
Křížení a mutace: mechanismus tvorby nových jedinců	Klonování a mutace	Tvorba alternativních rozvrhů

4.4.4.2 Mravenčí kolonie ACO (Ant Colony Optimization)

Tato metoda je inspirována výzkumem chování reálných mravenců [137], modelováním a simulacemi [138] jejich chování. Algoritmus ACO (Ant Colony Optimization), který je schopný optimalizovat cesty v grafech (TSP či JSSP) byl vyvinut poměrně nedávno roku 1996 [139]. Algoritmus pracuje na principu analogie mravenců hledajících jídlo. Analogie popisuje mravence, který jako první vyjde z mraveniště náhodným směrem, aby vyhledal potravu, a přitom vypouští feromon, který slouží jako sledovač cesty jak pro něj, tak pro ostatní mravence. Jakmile najde jídlo, vrací se zpátky do mraveniště. Cestou zpět opět vypouští feromon a tím zvyšuje intenzitu informace o prošlé cestě. Ostatní mravenci následují danou cestu s pravděpodobností danou intenzitou feromonu. Čím více mravenců nalezne potravu v okolí cesty, tím je intenzita feromonu větší. Feromon však časem vyprchává a tím se mění jeho intenzita, až úplně zmizí. V tu chvíli mravenci vyráží znovu náhodným směrem.

ACO je tedy metoda založená na vizualizaci grafů v několika krocích. Nejprve množina mravenců hledá cestu náhodně a úměrně kvalitě nalezenému řešení kladou feromon, tedy zvyšují pravděpodobnost nalezení řešení se stejnou cestou, ale nesledují ji přesně vzhledem k tomu, že jejich rozhodování na cestě je stále náhodné.

Tento princip lze popsat jako princip lokálního prohledávání (avšak s více počátečními řešeními) s náhodným způsobem prohledávání okolí a pamětí (tak jako tomu je u zakázaného prohledávání).

Pravděpodobnost (14), se kterou se mravenec rozhoduje v uzlu, je dána [40]:

$$P_{i,j} = \frac{[\tau_{i,j}]^\alpha [\eta_{i,j}]^\beta}{\sum [\tau_{i,j}]^\alpha [\eta_{i,j}]^\beta} \quad (14)$$

kde $\tau_{i,j}$ je množství feromonu (15) na hraně i,j mezi uzly

α je parametr kontroly vlivu na $\tau_{i,j}$ ($0 \leq \alpha$)

$\eta_{i,j}$ je požadavek na hranu

B je parametr kontroly vlivu na $\eta_{i,j}$ ($\beta \leq 1$)

Přepočet feromonu je pak

$$\tau_{i,j} = \rho \tau_{i,j}^* + \Delta \tau_{i,j} \quad (15)$$

Kde $\tau_{i,j}^*$ je množství feromonu na dané hraně

ρ je poměr vypařování feromonu

$\Delta \tau_{i,j}^k$ je množství naneseného feromonu. Nabývá hodnot $1/L_k$ (L_k je ohodnocení k -té mravenčí cesty (hodnota účelové funkce) v případě, že se mravenec pohybuje po hraně i,j , a 0 pokud ne.

Mezi další varianty ACO patří např. mravenčí kolonie s elitismem, kdy nejlepší řešení nanáší feromon v každé iteraci spolu s ostatními mravenci, či Max-Min pracující s maximálním množstvím nanášeného feromonu nebo Rank (žebříčkový) systém, kde jsou řešení rozvrstvena podle váhových hodnot pro každé řešení.

Ačkoli se ACO považuje za pokročilejší optimalizační přístup, jeho výsledky v oblasti rozvrhování nejsou v porovnání s ostatními metodami natolik kvalitní [140]. To je dáno tím, že princip ACO je problémově specifický a tudíž vhodnější pro úlohy TSP.

4.4.4.3 Rojení částic (Particle swarm optimization)

Rojení částic PSO (Particle swarm optimization), vyvinutá Eberhartem a Kennedym [141], používá analogii inteligence roje tj. biologicko sociálního systému jako je hejno ptáků, či ryb. V situaci, kdy hejno hledá jídlo, každý z jedinců se náhodně rozprchne do prohledávaného prostoru. Každý z jedinců má určitý stupeň svobody a náhodnosti ve svém pohybu, což umožňuje nalézt větší koncentrace jídla. Dříve či později nalezne jedinec jídlo a dle vzorce chování předává tuto informaci jedincům v hejnu.

Algoritmus PSO obsahuje roj částic (jedinců) v n -dimenzionálním prostoru G , kde každá částice p má pozici $p.g \in G \subseteq R^n$ a rychlost $p.v \in R^n$. Pozice $p.g$ odpovídá genotypu a ve většině případů PSO se jedná o celé řešení daného problému (vzhledem k tomu, že ve většině případů $p.x=p.g$ a problém řešení X je také R^n a $X=G$). Rychlost $p.v$ jedince p určuje, jakým směrem bude nadále probíhat prohledávání. V případě nastavení v na vysokou hodnotu se prohledávají náhodně co nejvzdálenější řešení. Pokud je hodnota v nízká, prohledává se důkladně prostor blízko stávajícího řešení.

V iniciační fázi PSO se náhodně určí rychlosti jednotlivých částic. V každém následujícím kroku se aktualizuje rychlost a posléze pozice. Každá částice p uchovává díky paměti své nejlepší místo. Sociální komunikaci pak zastává seznam sousedů $N(p)$, který obsahuje částice, které nejsou vzdálené od $p.g$ než o definovanou vzdálenost δ .

PSO tedy vytváří nové částice q , které jsou nahrazeny novými p obsahujícími genotyp $p.g$ (polohu) (16) a jeho vzdálenost $p.v$ v závislosti na učícím faktoru c_i a d_i . Odlišuje se přitom nejlepší známá poloha nejbližších sousedů $best(N(p))$ (17) a celé populace $best(Pop)$ (18).

$$q.g_i(t+1) = q.g_i(t) + q.v_i(t+1) \quad (16)$$

$$q.v_i = p.v_i + (random_u(0, c_i) * (best(p) * g_i - p.g_i)) + (random_u(0, d_i) * (best(N(p))) * g_i - p.g_i) \quad (17)$$

$$q.v_i = p.v_i + (random_u(0, c_i) * (best(p) * g_i - p.g_i)) + (random_u(0, d_i) * (best(Pop) * g_i - p.g_i) \quad (18)$$

Důležitým faktorem je tedy rychlost částic. V případě, že je rychlost příliš velká, jedinci se příliš od sebe vzdalují (a tím i od nejlepšího řešení).

V PSO existuje za krátkou existenci tohoto algoritmu velké množství modifikací[142]. Jednou z modifikací je definování sousedství pomocí sociálního příbuzenství, kdy o sousedství rozhoduje pořadí jedince v populaci. Tím se předchází k brzkému uváznutí jedinců v jednom z lokálních extrémů.

Další modifikací jsou druhové PSO jako ovčí SFHA [143] (Sheep Flock Heredity Algorithm), kde se části populace odlišují od ostatních. Zde jsou skupiny jedinců, kteří se mohou křížit pouze uvnitř skupiny a jsou tímto izolováni od ostatních. Během iterací jsou vybírány skupiny nejlepších jedinců, kteří ostatní ze své skupiny vedou (či se množí).

PSO na rozdíl od klasických evolučních algoritmů mají však tu nevýhodu [144], že brzo konvergují k lokálním extrémům a mají v některých případech v závislosti na typu PSO více nastavitelných parametrů (počet částic, definice velikosti prohledávaného prostoru, rychlost, učící faktory atd.) než u jiných algoritmů.

4.5 Dílčí závěr č.2 – Výběr metod pro tvorbu a optimalizaci dílenských rozvrhů

Většina reálných rozvrhovacích problémů je NP-těžká a tudíž komplexita a efektivnost algoritmu jsou jedny z kritérií při výběru metody pro aplikaci. Komplexita algoritmu zásadně ovlivňuje čas výpočtu a efektivita určuje kvalitu výsledků poskytovaných v čase.

Přes to, že algoritmy byly řazeny do jednotlivých podmnožin na přesné (např. metoda větví a mezí) a přibližné (konstruktivní algoritmy, metody založené na lokálním prohledávání, evoluční algoritmy, metody umělé inteligence), nelze jednoznačně tyto metody oddělit. To z toho důvodu, že některé metody mají vlastnosti z jiného okruhu metod. Příkladem metod založených na lokálním prohledávání může být simulované žihání či zakázané prohledávání, které přes to, že vykazuje typickou vlastnost lokálního prohledávání (prohledávání z jednoho řešení pomocí transformací tohoto řešení), vykazuje i vlastnosti typické pro metody umělé inteligence. Touto vlastností je paměť,

díky které přizpůsobuje svoje prohledávání a tím se de facto učí tak, jako tomu je např. u imunitních algoritmů. Podobně je tomu u metody rojení částic. Ačkoli vykazuje vlastnosti typické pro evoluční algoritmy, bývá často považována pro její schopnost učení za metodu spadající do třídy metod umělé inteligence.

Výhodou přesných metod (tab. 4-8) je jejich schopnost dosáhnout optimálního řešení, avšak s vysokým výpočetním časem. Pro řešení praktických rozvrhovacích problémů se nepoužívají, nicméně metoda větví a mezí se využívá pro nalezení optimálních řešení teoretických problémů pro následné testování schopnosti dosáhnout optima u přibližných metod.

Tab. 4-8: Přesné metody

Algoritmus	Charakteristika	Výpočet	Nejčastější aplikace v rozvrhování
Metoda větví a mezí	Konstrukce rozhodovacích stromů do hloubky či šířky.	Vždy nalezne optimum, výpočetní čas exponenciálně stoupá s lineárním růstem komplexity problému.	V praxi prakticky nevyužívané, aplikace možná na malé problémy či v případě teoretických problémů pro získání optimální hodnoty účelové
Matematické programování	Vysoce strukturované matematické modely, konstrukce algoritmu na míru daného problému.	Nižší než u metody větví a mezí, časové náklady na konstrukci problému jsou vysoké, vždy dosáhne optimální hodnoty účelové funkce.	Vzhledem k složitosti tvorby modelu (není univerzální) se prakticky nepoužívá.

Konstruktivní metody jsou v praxi nejvíce používané s ohledem na krátký čas, za který poskytují reálná řešení (tab. 4-9). Pravděpodobnost dosažení optimálních řešení v případě složitých rozvrhovacích problémů je malá. V oblasti teoretických problémů se využívají pro konstrukci navržených řešení.

Tab. 4-9: Přibližné metody – konstruktivní algoritmy

Algoritmus		Charakteristika	Výpočet	Nejčastější aplikace v rozvrhování
Konstruktivní algoritmy	Prioritní pravidla	Grafová reprezentace stromového problému pro aktivní rozvrh a rozvrh bez zpoždění, rozhoduje se v uzlech pomocí prioritních pravidel.	Nejrychlejší způsob tvorby rozvrhů, kvalita výsledků závisí na typu modelu a prioritním pravidlu, zřídka kdy dosáhne optimální hodnoty účelové funkce.	Nejčastější použití v praxi, používá se i jako prostředek výpočtu účelové funkce u pokročilých metod.
	Hrdlo lahve	Přerozvrhování na základě zpoždění na momentálním úzkém místě.	Pomocí určení úzkého místa (pracoviště s největšími nároky na výrobu). Přizpůsobení ostatních pracovišť úzkému místu. Dosahuje optimálních řešení v závislosti na počátečním řešení.	Užívá se jak v teoretických algoritmech (spíše jako část memetických algoritmů), tak v praktických aplikacích - systémy OPT
	Vkládací algoritmy	Vychází z částečných rozvrhů a vkládá nerozvržené operace.	Na základě náhodně vygenerovaných rozvrhů určité hloubky, operace se vkládají s ohledem na změnu účelové funkce. Dosahuje optimálních řešení v závislosti na náhodně generované počáteční řešení.	V oblasti teoretických rozvrhů se prakticky nepoužívají, jejich hlavní aplikací je v praxi v kombinaci s prioritními pravidly.

Metody založené na lokálním programování jsou aplikované v praxi vzácně. Metodu lokálního prohledávání využívají systémy OPT ve spojení s konstruktivní technikou hrdla lahve. V teoretické oblasti jsou však velmi využívány a to zejména pro schopnost dosáhnout optimálních řešení, avšak za vyšší výpočetní čas než u metod konstruktivních (tab. 4-10).

Tab. 4-10: Přibližné metody – metody založené na lokálním prohledávání

Algoritmus		Charakteristika	Výpočet	Nejčastější aplikace v rozvrhování
Lokální prohledávání	Horolezecký algoritmus	Prohledávací algoritmus. Možnost se vyhnout slabým lokálním extrémům.	Prohledávání s možností přijetí řešení s horší hodnotou účelové funkce. Malá časová náročnost a nízká pravděpodobnost nalezení optimálního řešení.	Pro svoji jednoduchost aplikovaný v testování algoritmů a etalonových modelů.
	Simulované žhání	Prohledávací algoritmus s dynamickou změnou velikosti prohledávané oblasti. Vykazuje známky algoritmů umělé inteligence – paměť.	Prohledávání se zvětšující a zmenšující se oblastí prohledávání dle modelu žhání krystalické mřížky kovů. Časová náročnost a pravděpodobnost nalezení optima úměrně počáteční teplotě a počtu zahřátí a shlazení.	Aplikuje se spíše v rámci jiných optimalizačních metod (memetické algoritmy).
	Zakázané prohledávání	Prohledávací algoritmus s krátkodobou a dlouhodobou pamětí. Omezená schopnost se učit.	Prohledávání stromu s uchováváním již použitých transformací. Dle nastavení parametrů možnost dosahovat optimálních hodnot účelové funkce. Nejpokročilejší prohledávací nástroj. Vysoké	Použití pro testování algoritmů a modelů v teoretické oblasti, v praxi se využívá v oblastech plánování projektů.

Evoluční algoritmy mají na rozdíl od metod lokálního prohledávání větší pravděpodobnost dosažení optima, ale jsou v závislosti na nastavených parametrech náročnější na čas výpočtu. V praxi se aplikují pouze vzácně a to zejména v oblasti ověřování na specifických problémech (tab. 4-11).

Tab. 4-11: Evoluční algoritmy

Algoritmus		Charakteristika	Výpočet	Nejčastější aplikace v rozvrhování
Evoluční algoritmy	Genetický algoritmus	Pravděpodobnostní prohledávání. Řešení reprezentovaná chromosomy.	Populace řešení je rekombinována pomocí selekce, křížení a mutace. Schopnost vyhnout se lokálním extrémům. Vysoká náročnost na výpočet.	Využití v celé řadě problémů, univerzálnost, omezené binární reprezentací. Přes to, že někteří výrobci ERP deklarují použití GA, nejsou známé reference.
	Genetické programování	Využití evolučních principů pro tvorbu software.	Populace je tvořena subprogramy řešící danou problematiku. Vysoká náročnost na výpočet. Schopnost dosáhnout globálních extrémů.	V oblasti rozvrhování se nepoužívá, testováno pouze na teoretických problémech. Využití v dataminingu.
	Evoluční strategie	Principy podobné GA, využití reprezentace reálných problémů.	Problémově orientované algoritmy, důraz kladen na mutaci a selekci. Schopnost dosáhnout globálních extrémů. Vysoká časová náročnost.	Využití v teoretických problémech. Testováno na praktických problémech.
	Memetické algoritmy	Hybridizace evolučních algoritmů principy transformací známých z prohledávacích metod.	Evoluční algoritmy, kde mutaci zastávají techniky lokálního prohledávání. Dosahují globálních extrémů. Vysoká časová náročnost.	Využití v teoretických problémech. Testováno na praktických problémech.

Metody založené na umělé inteligenci mají větší schopnost se vyhnout lokálním extrémům, avšak jejich časová náročnost bývá větší. Jejich aplikace v praxi podobně jako u EA bývá na úrovni pilotních projektů, kterých je v současnosti méně než u EA vzhledem k tomu, že tyto metody jsou poměrně nové (tab. 4-12).

Tab. 4-12: Metody umělé inteligence

Algoritmus		Charakteristika	Výpočet	Nejčastější aplikace v rozvrhování.
Umělá inteligence	Imunitní algoritmus	Evoluční principy obohacené o schopnosti učení. Velká schopnost vyhnout se lokálním extrémům.	Princip evolučních algoritmů a imunitních systémů, hodnocení jedinců na základě podobnosti, rozpoznávání funkcí genů. Schopnost dosáhnout globálních extrémů. Vysoká časová náročnost.	Využití v teoretických problémech.
	Mravenčí kolonie	Princip hmyzu hledající potravu spojená s komunikací jedinců.	Nejúspěšnější jedinci předávají informace o poloze (sekvenci) potravy (řešení). Nižší časová náročnost, náchylné ke sklouznutí do lokálního extrému.	Využití v teoretických problémech.
	Rojení částic	Široké spektrum metod jejichž vlastnosti závisí na chování organismů žijících v hejnech (stádech).	Založené na EA, princip předávání informací svým sousedům.	Využití v teoretických problémech.

Při volbě optimalizačních algoritmů lze vycházet z následujícího:

- Přesné metody jsou pro následnou analýzu nevhodné vzhledem k jejich časové náročnosti a aplikovatelnosti v praxi.
- Konstruktivní metody jsou nezbytným nástrojem, jak již v praxi tak v oblasti teoretického testování algoritmů.
- Metody lokálního prohledávání a zvláště jejich mechanismy transformace plánů mají značný přínos v oblasti rozvrhování.
- Evoluční algoritmy spolu s metodami umělé inteligence nabízejí obrovský potenciál a mohou, díky nalezení optimální plánů, poskytnout konkurenční výhodu.

Na základě provedené analýzy se jako vhodné ukazuje využití následujících optimalizačních metod:

- Konstruktivní algoritmus pro tvorbu rychlých rozvrhů pomocí prioritních pravidel.
- Metodu lokálního prohledávání, která optimalizuje/zpřesňuje řešení z konstruktivního algoritmu.
- Evoluční algoritmus pro tvorbu reálných plánů v přijatelném čase.

V následující kapitole je navržena metodika využívání heuristických algoritmů v rozvrhování výroby v SME, která vychází z dosavadních závěrů.

5 METODIKA VYUŽITÍ HEURISTICKÝCH ALGORITMŮ V ROZVRHOVÁNÍ VÝROBY V SME

Praktické využívání heuristických algoritmů v rozvrhování výroby je přes rozsáhlé znalosti zvláště v teoretické oblasti stále výjimečné. Přes to, že si všechny výrobní organizace uvědomují nezastupitelnou roli plánování a rozvrhování, stále používají ruční rozvrhování, ve většině případů pak systémy MRPI až MRPII (v rámci ERP systémů zaměřujících se na výrobu), jejichž principy lhůtového a kapacitního plánování jsou známy již z přelomu padesátých a šedesátých let dvacátého století.

5.1 Problematika zavádění informačních systémů řízení výroby

Prvními informačními systémy, které využívají principy rozvrhování (nepočítaje elektronické plánovací tabule), jsou pak APS a MES systémy. APS jako první využívají principy plánování do omezených kapacit resp. rozvrhování s tím, že hlavní důraz je kladen na samotná omezení (seřizování, směnnost, alternativní technologie) a parametry výrobních dávek (velikost výrobní dávky). Samotná oblast rozvrhování je pak řešena často pouze principem konstruktivních algoritmů. Prvním z APS systémů soustředujícím se na tvorbu a optimalizaci rozvrhů byly pak OPT systémy, které využívají principy konstruktivního algoritmu hrdla lahve, jež konstruuje rozvrh pro kritická pracoviště (úzká místa či hrdla lahve) a odvozuje od tohoto rozvrhu lhůtový plán pro pracoviště nekritická. Optimalizace rozvrhu je pak prováděna změnou sekvence zakázek na kritických pracovištích.

MES systémy přes to, že jejich hlavní funkcí je poskytování zpětné vazby o stavu výroby, jsou progresivním nástrojem v oblasti dílenského rozvrhování. Jejich hlavní funkcí bude v budoucnu optimalizace rozvrhu s ohledem na momentální stav ve výrobě. Řada MES již má tuto funkcionalitu, ale stále pracují zejména na úrovni tvorby rozvrhů pomocí konstruktivních pravidel či ručního rozvrhování (systémem „drag and drop“), které může bez znalosti komplexních vazeb v rámci rozvrhu způsobit destrukci celého plánu (rozvrhu).

Posledními nástroji, které díky důvěře získané v oblasti projektování výrobních systémů pronikají do praxe v oblasti rozvrhování, jsou klasické simulační nástroje [145, 146]. Simulační nástroje s funkčním rozvrhováním jsou však v dnešní době jen málo dosažitelné pro podniky SME vzhledem k tomu, že pro jejich uspokojující funkcionalitu je potřebné napojení na ERP systémy, jejichž cena je stále ještě pro většinu SME nedosažitelná.

Implementace pokročilých principů rozvrhování resp. jejich nástrojů je v praxi odmítána zvláště z důvodů negativních zkušeností s implementacemi APS systémů a to z několika důvodů:

- Představa, že rozvrhovací nástroj nahradí celý ERP systém – ERP systém nelze nahradit, jelikož obsahuje vstupní data pro rozvrhovací nástroj (technologické postupy, informace o výrobním systému – stroje, informace o omezeních – směnnost, atd.).
- Neexistence vstupních dat pro rozvrhování či jejich nekonzistence - bez vstupních dat je rozvrhovací systém pouze drahou vizitkou společnosti. Rozvrhování pomocí nesprávných vstupních dat poskytuje pouze nesprávné rozvrhy.

- Rozvrhování není samospasitelná metoda – tak jako všechny metody je určena pro určité modely výroby. Typickou oblastí, kde je rozvrhování bezúčelné, je velkosériová výroba jednoho typu výrobku na sklad dle prognózy.
- Očekávání, že rozvrhovací nástroj sám vyřeší všechny problémy – je nutné si uvědomit, že se jedná pouze o nástroje, které pomáhají lidské činnosti automatizací podle předem nastavených parametrů. Pokročilé nástroje jsou schopny se přizpůsobovat měnícím se podmínkám a učit se. Přesto má stále hlavní slovo rozhodnutí člověka. Tyto systémy mu práci pouze ulehčují.

Motivací pro rozvrhování pak může být několik. Zpravidla to bývá nutnost začít plánovat do omezených kapacit spolu se sledováním začátku a konce jednotlivých operací.

V tomto kroku je důležité si uvědomit nejednoznačnost v pojmu plánování do omezených kapacit. V řadě případů se totiž nejedná o rozvrhování, ale pouze o pokročilé plánování, kdy plánovací algoritmus zabezpečuje, že náročnost práce nepřekročí 100% plánované kapacity.

To bývá dostatečné v těchto případech:

- Výroba typu linka, kde je jedno stabilní úzké místo – v takovém případě je nutné kontrolovat nepřesáhnutí kapacitních požadavků, popřípadě detailně rozvrhovat úzké místo. Zbýlá pracoviště se plánují „pouze“ lhůtově s ohledem na rozvrh úzkého místa.
- Vybalancovaná linka či výrobní buňka se stabilním zákaznickým požadavkem – takovýto typ výroby je obvykle řízen metodou kanban (samo-řídící regulační okruhy) či Heijunka (nivelizovaná výroba). Plánování pomocí Heijunka bývá často považováno mylně za rozvrhování. Sice se jedná o určování sekvence, ne však začátku a konce operací.

Tyto způsoby plánování jsou v současné době implementované i v oblastech spíše vhodných pro rozvrhování výrobních dávek, a to díky jejich použití v automobilovém průmyslu, který pro mnohé organizace slouží jako vzor.

Základní znaky výrobního systému, u kterého je vhodné implementovat rozvrhovací metody, jsou:

- Zakázková - kusová či malo-sériová výroba.
- Výroba širokého spektra výrobků s rozdílnou trajektorií výrobním systémem.
- V technologickém postupu se výrobky často vrací na pracoviště, na kterém již byly zpracovávány.
- Přes to, že pracoviště kapacitně vyhovují, často vznikají požadavky na zpracování různých výrobků ve stejný čas, a tím dochází ke zpoždění zakázek.
- Ke splnění zákaznického požadavku je potřeba malé množství dlouho trvajících operací.
- Operace lze zcela či částečně provést na jiných pracovištích (pracoviště lze pro danou operaci zaměnit).

Přes to, že výrobní systém vykazuje tyto vlastnosti, rozvrhování a optimalizace rozvrhu nemusí mít žádaný efekt. Řada společností v minulosti řešila rozvrhovací

problém nákupem více zařízení, ačkoli vytížení pracovišť nedosahovalo 100%. V takových případech docházelo k eliminaci rozvrhovacího problému vytvořením improvizovaných linek pro každý z typů dílů.

Postup při rozhodování, zda a jakým způsobem implementovat nástroj využívající heuristické metody rozvrhování by tedy měl obsahovat následující kroky:

- Definování požadavků na plán – rozvrh
- Stanovení měřitelných ukazatelů
- Analýza současného stavu
- Testování modelu pomocí navržených rozvrhovacích metod a rozhodnutí o vhodnosti implementace.

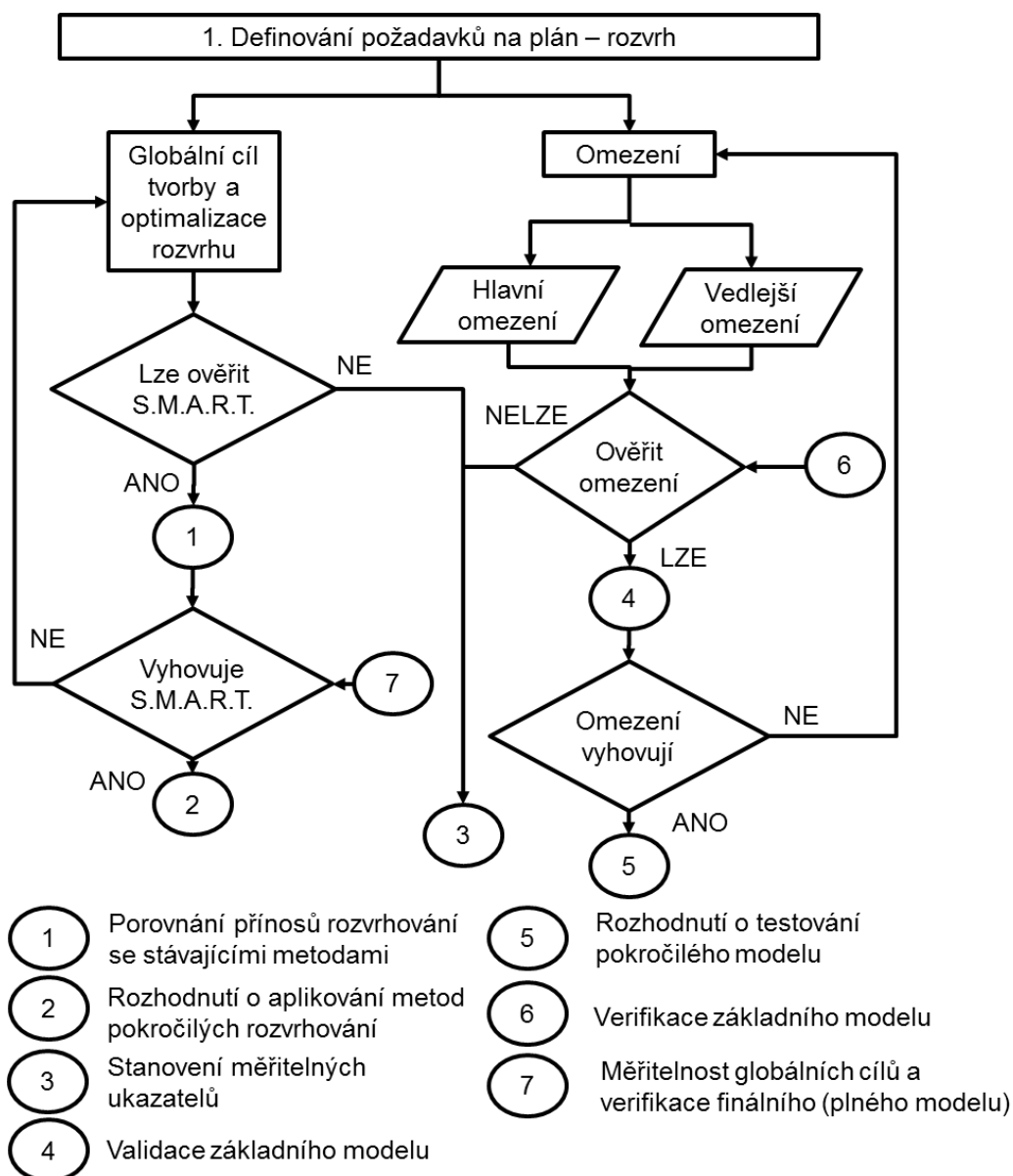
5.2 Definování požadavků na plán – rozvrh

Ve fázi definování požadavků na plán - rozvrh je nutné určit všechny základní atributy, které ovlivňují zadání rozvrhu a jeho finální výstup. Hlavními atributy jsou kritéria hodnocení plánu - rozvrhu, neboli definování účelových funkcí (více viz kapitola 4.1) a zdroje, resp. omezení definující naplnění kritérií hodnocení.

Jak je patrné z obrázku 5-1 je definování požadavků na rozvrhování složitým krokem, u kterého lze předpokládat, že jeho náplň (cíl a omezení) se v průběhu projektu posouzení aplikování rozvrhovacích metod mění. Definování globálního cíle je silně provázáno s následujícími kroky (spojovacími body):

- 1) Porovnání přínosů rozvrhování se stávajícími metodami.
- 2) Rozhodnutí o aplikování metod pokročilého plánování (rozvrhování).
- 3) Stanovení měřitelných ukazatelů.
- 4) Validace základního modelu.
- 5) Rozhodnutí o testování pokročilého modelu.
- 6) Verifikace základního modelu.
- 7) Měřitelnost globálních cílů a verifikace finálního (plného modelu).

V rámci definování globálních cílů zavádění rozvrhovacích metod je nutné brát ohled na navazující krok stanovení měřitelných ukazatelů resp. účelových funkcí vyjadřujících naplnění těchto cílů.



Obr. 5-1: Postup při definici požadavků na plán rozvrh

Při definování zdrojů a omezení se definují základní zdroje, které se budou rozvrhovat. Často se pojem zdroje a omezení sjednocují (ne však v teorii omezení). Zdroj lze chápat jako primární omezení, jehož dostupnost je v rozvrhovací sekvenci ověřována jako první. Omezení pak upravuje dostupnost zpracovávání operace. Princip ověřování dostupnosti jak u zdroje, tak u omezení bývá totožný. V některých případech se zdroj ověřuje v časovém oknu (rozvrhuje), omezení se pak pouze plánuje kapacitně. Za zdroje a omezení lze považovat:

- 1) Materiál – jedná se o zdroj, který definuje termín uvolnění zakázky do výroby. Řízení tohoto zdroje však nebývá zahrnuto v rozvrhování, řídí se pomocí MRPI.
- 2) Stroj – jedná se o nejčastější zdroj.
- 3) Přípravek – spíše uvažován jako omezení, avšak v případě výroby výlisků se pokládá za zdroj (např. více lisů o stejné pracovní síle a pracovním prostoru však s omezeným množstvím lisovacích forem).

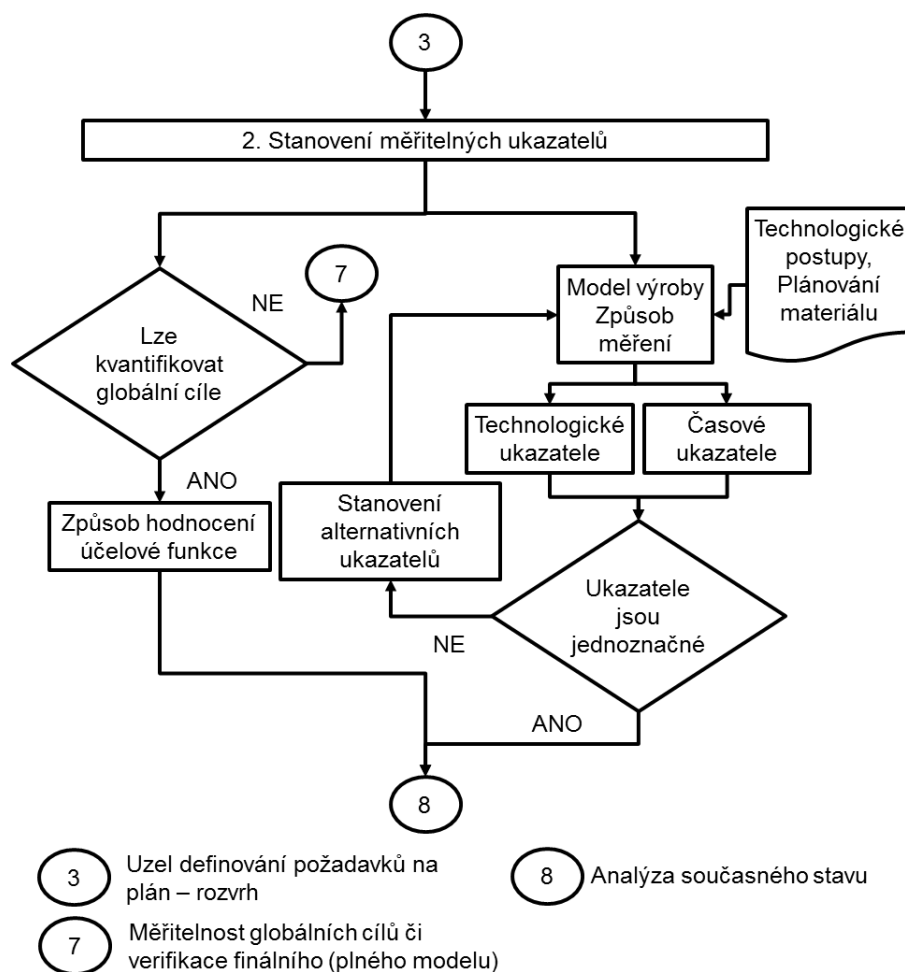
- 4) Pracovník - pracovníci se zpravidla uvažují zejména jako dodatečná omezení. Existují však případy (zejména ve flexibilních výrobních buňkách), kdy je pracovník hlavním zdrojem. V těchto případech se však místo klasických rozvrhovacích metod volí spíše metody zaměřené na organizaci práce spojené s co – když (what - if) simulací. Mezi ně řadíme např. rozvrhovací diagramy (SD – Scheduling diagrams), tok jednoho kusu – karavana (OPFC - One Piece Flow Caravan) či hašení požáru (BB – Bucked Brigades) [147].
- 5) Směnnost – zpravidla se jedná o omezení používané ve výrobních systémech s různou směnností (8 hodinová a 12 hodinová směnnost).
- 6) Transport a kooperace – omezení, které se nejčastěji plánuje kapacitně s tím, že pouze oddaluje zpracování zakázky na nadcházejícím rozvrhovaném stroji. Existují však výjimky jako například rozvrhování manipulační techniky v konsignačních skladech.
- 7) Poruchy a zmetky – jedná se o omezení, které je nejčastěji řešeno kapacitně s ohledem na výskyt poruch.
- 8) Rozvrhovací horizont – nejkratší plánovací horizont lze zvolit na základě frekvence odvádění hotových výrobků (např. jedna hodina či směna), nejdelší (např. den až měsíc) pak s ohledem na četnost nutných zásahů do modelu danou zkušeností plánovačů a vlastnostmi vytvořeného modelu výroby. Rozvrhovací horizont v závislosti na počtu operací a jejich délky ovlivňuje dobu výpočtu rozvrhu.

Dalších omezení je celá řada (například zákazník, skladovací prostory atd.), avšak nejpoužívanějším zdrojem (omezením) je dostupnost materiálu na zpracování a stroj, na kterém se tento materiál zpracovává.

Další krok obsahuje kvantifikování měřených omezení a definovaných cílů.

5.3 Stanovení měřitelných ukazatelů

V rámci kroku stanovení měřitelných ukazatelů se vychází z předchozího kroku definování požadavků na plán. Tyto se kvantifikují s ohledem na sledování těchto ukazatelů v praxi (obr. 5-2).



Obr. 5-2: Postup stanovení měřitelných ukazatelů

Měřitelné ukazatele pak lze rozdělit dle:

- 1) Účelová funkce/více funkcí – ve většině případů se jedná o časová hlediska, které jsou jednoduchým způsobem měřitelná. Více účelových funkcí lze nejjednodušším způsobem kvantifikovat (termín odvedení všech zakázek, využití pracovišť, brzké odvedení či zpoždění zakázky vůči stanovenému termínu atd.) pomocí ekonomického hlediska. To je však často obtížné vzhledem k neexistenci takových údajů. Při nejednoznačnosti hodnocení rozvrhů je nutné pak přistoupit k jejich revizi v rámci globálních cílů (obr. 5-2 spojovací uzel č. 7).
- 2) Ukazatele modelu výroby:
 - I. Technologické časy – stanovení nejmenší časové jednotky (sekundy, minuty, hodiny) pro technologické údaje (kusový čas, dávkový čas atd.). Tyto jednotky je nutné dodržet v celém systému i v rámci omezení (směnnost).

- II. Jednotky materiálu a polotovarů – stanovení řezných plánů či poměrového převodu, kdy je obvyklé, že materiál se ve strojírenství nakupuje v hmotnostních jednotkách a hotové výrobky či polotovary v kusech.

Důležitým prvkem tohoto kroku je stanovení způsobu měření ukazatelů. V případě časů technologických operací se často vychází u kusového času ze simulace CNC (Computer Numeric Control) kódu, či metod měření času. U časů seřizovacích pak lze využít monitorování seřízení z MES systému, či využití klasických analytických metod jako je MTM (Methods Time Measurement) a MOST (Maynard Operation Sequence Technique). U ostatních se obvykle vychází ze statistických dat pořízených např. časovým snímkem dne.

Po stanovení měřitelných ukazatelů se přistupuje k analýze současného stavu a tvorbě základních prvků modelu (obr. 5-2 spojovací uzel č. 8).

5.4 Analýza současného stavu

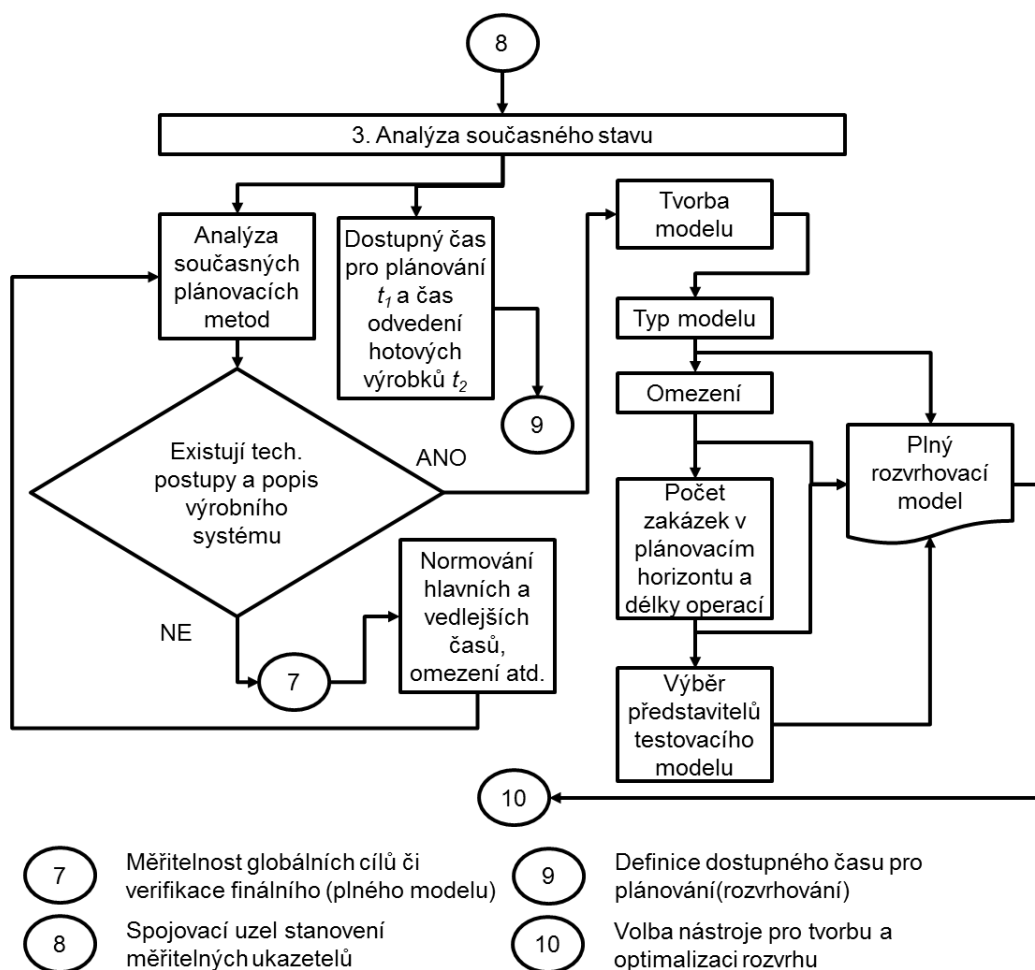
Analýza současného stavu bývá časově nejnáročnějším krokem i s ohledem na časté vracení se do předchozích částí v případech, kdy se díky analýze zjistí, že měřitelné ukazatele a ve výjimečných případech i definování požadavků bylo nevhodně zvoleno.

V případě analýzy současného stavu se s ohledem na zavádění principů rozvrhování pomocí heuristických metod klade důraz především na:

- 3) Výběr hlavních představitelů – představitele pro modelování výrobního systému lze vybírat dle několika hledisek. Prvním hlediskem je důležitost daného výrobku dle ekonomických ukazatelů (výběr např. pomocí ABC analýzy). Druhým hlediskem je pak složitost výrobní trajektorie výrobku systémem, resp. složitost technologického postupu. Tím bývá zásadně ovlivněna efektivnost metod. Čím je výrobní postup z tohoto pohledu složitější (např. vracení se dílů na stroje, kde již byly zpracovávány v některé z předcházejících operací), tím dochází k větší kombinatorické složitosti modelu a tím je také větší pravděpodobnost nalezení lepšího řešení pomocí pokročilejších metod, než u např. prioritních pravidel. Na tomto základě lze pak v následujících krocích rozhodnout o nasazení metod.
- 4) Druh výroby – v rámci tohoto bodu se analyzuje technologický postup s ohledem na podobnost k některému z teoretických modelů. Jedná se o základní rozdělení na FSSP, JSSP či OSSP a jejich paralelní a flexibilní modifikace.
- 5) Množství zakázek či operací a délka trvání operací – čím větší množství zakázek (rozvrhovaných operací) a kratší doba trvání zpracovávání operací, tím je efektivnost pokročilých rozvrhovacích nástrojů menší [148]. Na druhou stranu příliš malé množství operací může v některých případech omezit kombinatorickou složitost rozvrhu a tím i přínosy pokročilejších metod rozvrhování.
- 6) Konzistence dat – je jedním z nejzásadnějších a neproblematictějších bodů. V případě, že jsou pro výpočet na vstupu chybná data, budou chybná data i na výstupu. Správné určení procesních a seřizovacích časů bývá klíčové, ale často se opomíjejí časy vedlejší, jako transport či kontrola kvality. Při nesprávném určení těchto časů lze v některých případech posoudit vhodnost nasazení heuristických metod rozvrhování, avšak řízení výroby pomocí takto vygenerovaných rozvrhů není vhodné. Nejčastější chyby se pak vyskytují v rámci:

- I. Identifikace zakázky.
 - II. Datum uvolnění zakázky.
 - III. Termín odvedení zakázky.
 - IV. Množství (požadavek a výrobní dávka).
 - V. Identifikace zdrojů (strojů) či omezení spolu s jejich kalendářem směn a typem stroje (zaměnitelný stroj, dávkový stroj jako např. lisy či „kapacitní stroj jako je žihací pec).
 - VI. Kusovníková struktura výrobku.
 - VII. Časy procesní, seřizovací, transportní.
- 7) Omezení výroby – úzce souvisí s konzistencí dat. V tomto případě je nutné prověřit stavy omezujících podmínek, jako jsou:
- Stroje - jak se bude realizovat jejich rozvrhování, pokud např. vyrábějí hromadně a ne po kusech.
 - Přípravky - jsou sdílené více stroji či ne.
 - Nástroje či pracovníci - jsou zdrojem nebo vůbec nepatří mezi omezující podmínky.
- V tomto kroku je vhodné posoudit, na kolik tyto zdroje jsou omezující v rámci rozvrhování a jestli je dostatečné množství a kvalita dat, aby je bylo možné použít pro rozvrhování.
- 8) Existující plánovací metody – ovlivňují jak samotnou tvorbu modelu, tak testování heuristických metod. Při tvorbě testovacích modelů je vhodné rozlišit výrobky vyráběné na sklad a zakázkové položky. Při testování modelu lze v rámci simulace porovnat předchozí přístup (v případě, že je ve výrobním systému aplikována některá z plánovacích metod) s výsledky optimalizačních algoritmů.
- 9) Způsob odvádění hotové výroby – zásadně ovlivňuje princip rozvrhování a rozhodnutí, zda optimalizovat rozvrhy. V případě, že by v rámci testování byla např. redukce času potřebného na zpracování zakázky nižší než interval odvádění výroby, není obvykle vhodné zavádět metody pokročilého rozvrhování.
- 10) Časová náročnost plánování a rozvrhování – analýza minimálních a maximálních časů dostupných pro fázi tvorby rozvrhu zásadně ovlivňují kvalitu získaného řešení (čím delší čas optimalizace, tím větší pravděpodobnost nalezení optimálního řešení).

Na obrázku 5-3 je popsána návaznost oblastí analýzy současného stavu, kde uzel č. 9 poskytuje informaci o dostupném času pro plánování (rozvrhování), která zásadním způsobem ovlivňuje volbu nástroje pro tvorbu a optimalizaci rozvrhu pomocí modelu výroby (uzel č. 10).



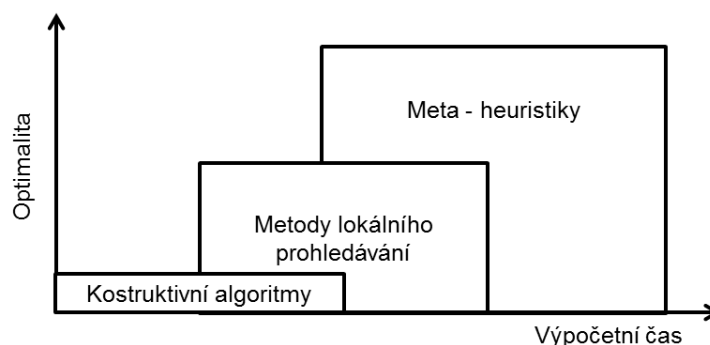
Obr. 5-3: Analýza současného stavu

Na základě těchto analýz se vytvoří model pro aplikaci rozvrhovacích metod. Tak jako v předešlých krocích se předpokládá, že v průběhu testování je pravděpodobné, že vzniknou nové informace, které mohou zásadním způsobem ovlivnit pohled na tvorbu modelu (zvláště v případě omezení) a je tedy nutné se vrátit k analýze dat a s tím spojené tvorbě nového modelu.

Dalším krokem při posouzení vhodnosti aplikace heuristických metod je tedy jejich vícefázové testování na modelu posuzovaného výrobního systému.

5.5 Testování modelu pomocí navržených rozvrhovacích metod

Při vybírání metod při optimalizaci modelu je třeba reflektovat stávající metody plánování a rozvrhování zvláště s ohledem na dostupný čas na optimalizaci a tudíž i potenciální kvalitu rozvrhu danou jednou nebo více účelovými funkcemi (obr. 5-4).



Obr. 5-4: Optimalita a výpočetní čas heuristických metod

Prvním krokem v obecném postupu tvorby a optimalizace rozvrhu výroby heuristickými metodami je vhodné zvolit konstruktivní algoritmus (CA), který, jak již bylo zmíněno, generuje rozvrhy rychle, avšak jejich kvalita (určeno pomocí hodnoty účelové funkce) bývá nižší než u ostatních heuristických metod.

Takto vygenerovaná řešení lze použít jako počáteční řešení pro některou z metod lokálního prohledávání, které jsou časově náročnější, avšak lze předpokládat, že naleznou lepší řešení (bližší optimu).

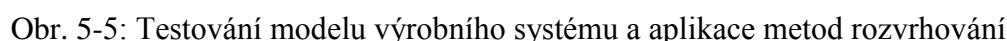
Nejlepších výsledků pak lze dosáhnout pokročilými meta-heuristikami, jako jsou evoluční algoritmy. Jejich aplikace závisí zejména na dostupném čase pro optimalizaci a kombinatorické složitosti rozvrhovacího problému reprezentovaného modelem.

V rámci testování navrženého modelu je vhodné postupovat v několika fázích s ohledem na složitost modelu a nadefinované časové požadavky na rozvrh.

Je vhodné rozdělit celý model na 2-3 modely a postupně je testovat:

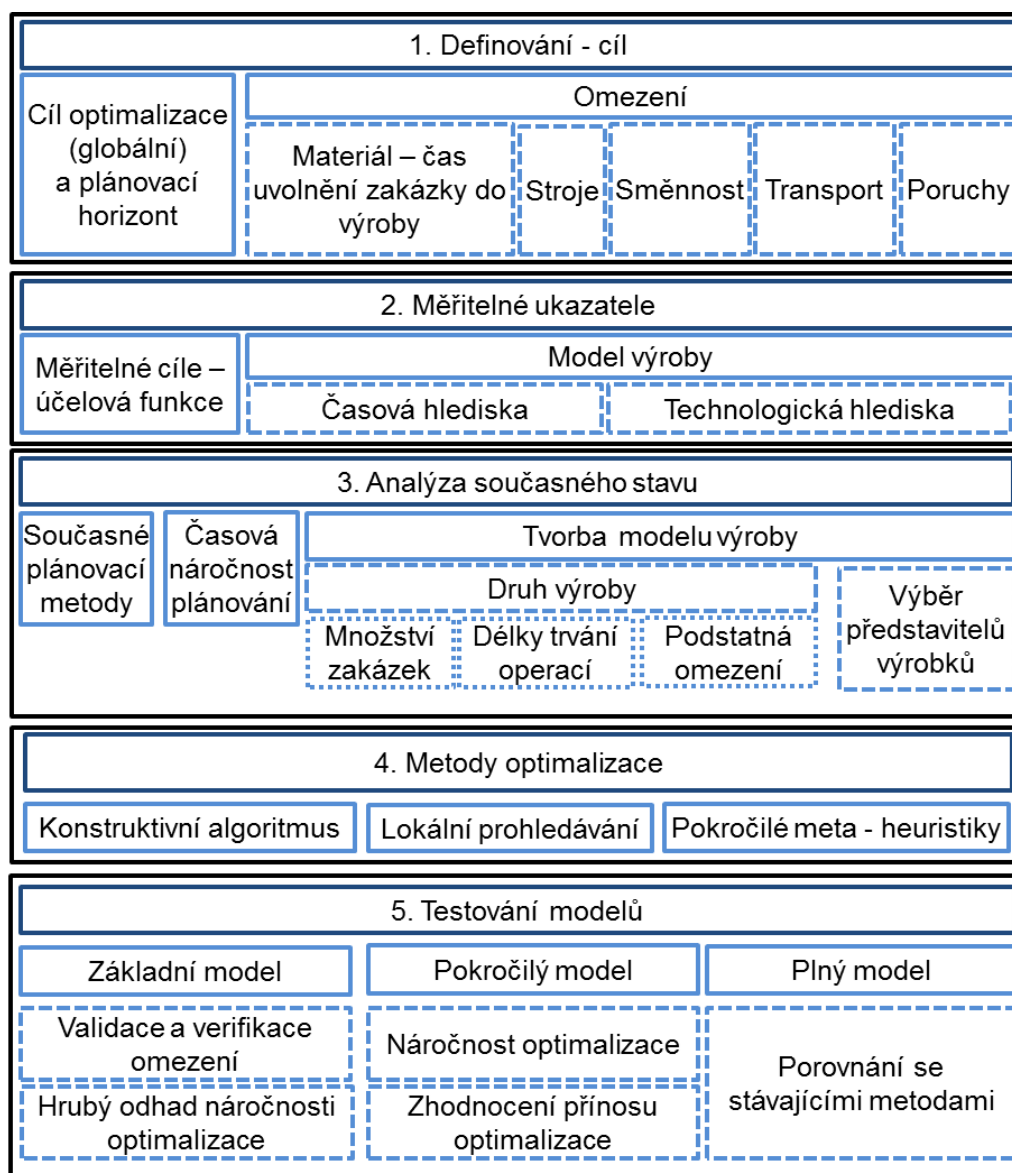
- 1) Testování prvního nejmenšího modelu (cca. 1/3 zakázek) obsahuje analýzu správného nastavení konstruktivního algoritmu, zvláště pak řízení logiky omezení. Zde se také analyzuje závislost počtu operací nutných k rozvržení modelu na výpočetním čase. Takto získaný čas je pouze orientační, obvykle v něm není zohledněna kombinatorická složitost. V případě, že na základě takového testování je ověřena konzistence dat a schopnost modelu poskytnout reálné rozvrhy, lze přistoupit k druhému kroku.
- 2) Po přidání zhruba druhé třetiny modelu lze v naprosté většině případů testovat časovou náročnost tvorby rozvrhů s ohledem na kombinatorickou složitost. V tomto kroku lze také testovat možný přínos optimalizace pomocí pokročilých metod a to ve dvou krocích - testování modelu s a bez termínu náběhu zakázek do výroby daným plánováním materiálu (MRPI):
 - I. Při testování modelu bez vazby na termín uvolnění materiálu do výroby lze odhadnout přínosy optimalizačních metod v závislosti na čase a nastavených parametrech optimalizačních metod. V tomto kroku lze odhadnout kombinatorickou složitost modelu. Přesné ověřování složitosti vyžaduje např. metodu větví a mezí, což je časově velmi náročné i s ohledem na potřebu analyzovat model při jeho změně.

3) Poslední model obsahující celý plánovací horizont slouží k ověření schopnosti pokročilých optimalizačních metod dosáhnout lepších výsledků a to za reálných podmínek simulovaných pomocí reálného modelu výroby zvláště s ohledem na frekvenci odvádění hotových výrobků (v případě účelové funkce makespan či zpoždění zakázek). V případě, že frekvence odvádění hotové výroby je vyšší než poskytnuté zlepšení účelové funkce lze konstatovat (obr. 5-5), že pokročilé metody rozvrhování není efektivní zavádět. Jedná se zejména o případy nepřetržité výroby. Hlavním přínosem rozvrhování je pak „plánování do omezených kapacit“ pomocí konstruktivního algoritmu a prioritních pravidel.



5.6 Dílčí závěr č.3 - Shrnutí metodického postupu

Podobně jako u implementace APS či MES systémů, v oblasti plánování (rozvrhování) výroby je nutné brát ohled na celý výrobní systém a to nejen na strojní zařízení, ale také na pracovníky a stávající procesy plánování. K tomu může napomoci v předchozích krocích nadefinovaná metodika, jejíž kroky vycházejí z obecných pravidel zavádění metod průmyslového inženýrství a projektového řízení. Tyto metody se často prolínají či doplňují, stejně jako metodiky DMAIC, PDCA či SMART. Ve všech případech je nutné dobře nastavit cíle, umět je kvantifikovat a mít relevantní data k jejich dosažení. Cíle je nutné řídit nejen dle ambiciózních představ, ale je třeba brát ohled i na současný stav výrobního systému. Nedílnou součástí je i návrh variant řešení a v případě, že ani jedna z variant není akceptovatelná, změnit přístup k řešení problému či přizpůsobit své cíle. Tak jako jiné metodiky aplikované ve výrobních systémech se i tato metodika snaží těmito pravidly řídit (obr. 5-6).



Obr. 5-6: Kroky zavedení rozvrhovacích metod

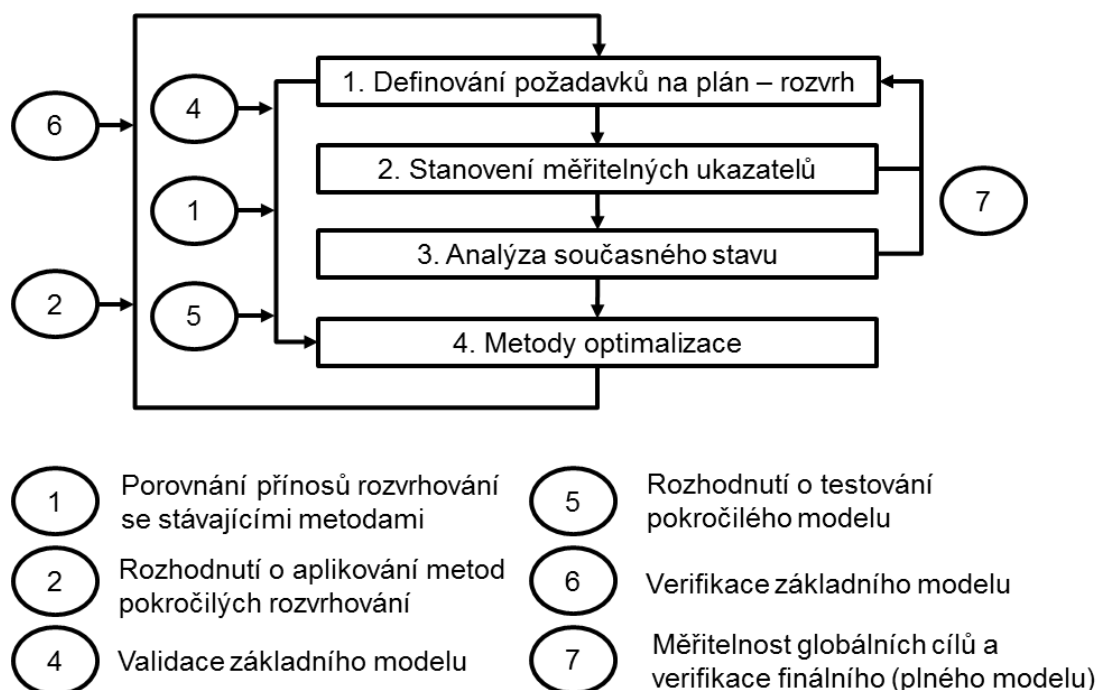
Průběh posouzení implementace rozvrhovacích metod (obr. 5-7) vychází z definice cíle (cílů), kde jednotlivé kroky vycházejí z předpokladů, které se mohou v

průběhu změnit. Jedná se zvláště o kroky stanovení měřitelných ukazatelů, kdy např. cíl snížení nákladů za opožděné dodání nelze kvantifikovat. Penále za pozdní dodání jsou sice obvykle kvantifikovaná, ale odhad ztrát na budoucích zakázkách vlivem ztrát zákazníků je založen opět na odhadu, a tak se přistupuje k redukci cíle na snížení času pozdního dodání (obr. 5-7 uzel č. 7).

V rámci analýzy současného stavu pak často dochází k opětovnému definování omezení aplikovaných v rozvrhovacím modelu (obr. 5-7 uzel č. 7), zvláště ve fázi testování modelu tzn. validace a verifikace omezení (obr. 5-7 uzel č. 4, 5 a 6).

Pokud model dostatečně popisuje současný problém v oblasti rozvrhování, lze hodnotit zvolené cíle z hlediska reálnosti a následně je přizpůsobit tak, aby byly akceptovatelné. To zvláště v případě, kdy frekvence resp. časový úsek, ve kterém je odváděna výroba, je delší, či čas investovaný do optimalizace je vyšší než čas uspořený (obr. 5-7 uzel č. 2). V takových případech je rovněž vhodné zvážit, zda investice do implementace rozvrhování v rámci APS, MES, či samostatného rozvrhovacího nástroje je vhodná. To vše s vědomím, že hlavním cílem, či přínosem takové implementace, je:

- znalost o možných konfliktech zakázek,
- zkrácení času rozhodování, které zakázce dát prioritu,
- přesnější odhad termínu dovedení zakázek díky rozvrhování resp. plánování do neomezených kapacit.



Obr. 5-7: Hlavní vazby kroků navržené metodiky

V následující kapitole jsou navrženy a testovány metody pro optimalizace rozvrhů.

6 NÁVRH METOD PRO TVORBU A OPTIMALIZACI DÍLENSKÝCH ROZVRHŮ

Tvorba metod v rámci rozvrhovacího nástroje se zabývá, tak jak bylo navrženo v kapitole 3, JSSP modelem a jeho modifikacemi. Metody aplikované na tento model byly pak vybrány v rámci kapitoly 4. Protože je kladen důraz zejména na krátkou dobu tvorby rozvrhů, je použit v první řadě konstruktivní algoritmus. Dále jsou testovány pokročilé optimalizační techniky, jako je metoda lokálního prohledávání, evoluční algoritmus a hybridní evoluční algoritmus. U těchto metod se předpokládá lepší hodnota účelové funkce, avšak delší časová náročnost. Testování se provádí na PC sestavách - CPU 3.2 GHz a 8GB RAM.

6.1 Konstruktivní algoritmus

Pro navrhovanou aplikaci byl jako základ konstruktivního algoritmu zvolen přístup reprezentace síťového grafu pomocí algoritmu Gifflera a Thompsona [101], který zaručuje tvorbu aktivního rozvrhu a jeho modifikace pro tvorbu rozvrhu bez zpoždění. V tomto algoritmu se rozvrhuje operace po operaci. Operace se považuje za rozvrhovatelnou v případě, že všechny operace, které jí předchází v technologickém postupu, jsou již rozvrženy. V případě teoretických JSSP problémů, kde je n zakázek a m strojů se pak jedná o nm iterativních kroků. Algoritmus generující aktivní rozvrhy (pro rozvrhy bez zpoždění jsou uvedeny podmínky v závorkách) rozvrhuje operaci v kroku t kde:

P_t – částečný rozvrh $(t-1)$ rozvržených operací.

S_t – množina operací, které lze rozvrhovat v kroku t , tzn. všechny operace, které předchází operacím v S_t , jsou v P_t .

e_t – nejdřívější čas, kdy operace o_k v S_t může začít.

f_t – nejdřívější čas, kdy operace o_k v S_t může skončit $f_t = o_k + p_k$, kde p_k je procesní čas operace o_k .

1. Necht' $t = 1$ s prázdnou množinou P_t . S_t je pak množina všech operací bez předchůdců, tzn. všechny počáteční operace zakázek
2. Nalezni $f_t^* = \min_{o_k \in S_t} \{f_t\}$ (rozvrh bez zpoždění - $e_t^* = \min_{o_k \in S_t} \{e_t\}$) a stroj M^* , na kterém se nalézá f_t^* (v případě rozvrhu bez zpoždění e_t^*). V případě, že takových strojů je více, vyber M^* náhodně.
3. Vyber množinu operací O (takzvaná množina konfliktních operací vyžadující stejný zdroj za stejných podmínek např. začátek operace) z S_t , které:
 - I. Vyžadují stroj M^* .
 - II. $e_t < f_t^*$ (u rozvrhu bez zpoždění - $e_t^* = e_t$)
4. Vyber vhodnou o_j z O
5. Posun na další krok:
 - I. Přidání o_j do P_t tzn. vytvoření P_{t+1}
 - II. Smazání o_j z S_t a vytvoření S_{t+1} přidáním operace, která bezprostředně následuje za o_j v dané zakázce (úloze), do S_t . Do S_t se nepřidává další operace, pokud byla o_j poslední v dané zakázce.
 - III. Zvětšení t o 1
6. V případě, kdy jsou některé operace nezaplánovány (pro teoretické problémy $t \leq nm$), tak se pokračuje na krok č. 2, jinak ukonči rozvrhování.

Aktivní rozvrh je použit v dalším výzkumu vzhledem k možnosti dosáhnout optima a rozvrh bez zpoždění vzhledem k menšímu prohledávacímu okolí, tj. kratšímu času nutnému k optimalizaci. Oba tyto algoritmy jsou taktéž blízké svým přístupem algoritmům, které využívají pro rozvrhování ERP/APS systémy.

Jedním z nejčastějších omezení ve výrobních zakázkových typu bývá možnost zpracování operace na více strojích. Je tomu tak v případech, kdy výrobní systém je uspořádán technologicky (frézovna, lisovna atd.), zejména pak, když je možné využít počítačem řízených strojů (CNC, obráběcí centra atd.) vybavených víceúčelovými nástroji.

Základní rozvrhovací problém určení pořadí zakázek před jednotlivými stroji je pak rozšířen o přiřazovací problém. Zde se vybírá jeden stroj z množiny strojů schopných zpracovat danou operaci. Takovýto rozvrhovací model se označuje jako FJSSP (viz kapitola 3.2.4).

Konstruktivní algoritmus FJSSP se od modelu JSSP příliš neliší. Při výběru konfliktní množiny operací se uvažují všechny možnosti, kde lze operaci rozvrhnout dle klíče daného druhem rozvrhu – u aktivního $f_i^* = \min o_k \text{ v } S_k \{f_i\}$ u rozvrhu bez zpoždění - $e_i^* = \min o_k \text{ v } S_k \{e_i\}$. Výběr operací do konfliktní množiny se pak neomezuje pouze na stroj M^* , ale na všechny operace, které v případě aktivního rozvrhu splňují podmínku $e_i < f_i^*$ a u rozvrhu bez zpoždění $e_i^* = e_i$.

Nejtěžším krokem popsaného algoritmu je pak krok č.4, tzn. vybrání vhodné operace o_j z O z množiny konfliktních operací na stroji M^* . K tomu slouží tzv. prioritní pravidla (kapitola 4.4.1). Pro testování tohoto algoritmu byla vybrána pravidla:

- SPT (Shorter Processing Time) – zakázka s nejkratším procesním časem a to vzhledem k častému výskytu tohoto pravidla v rozvrhovacích systémech. Jedná se o pravidlo, které patří mezi ta nejjednodušší vzhledem k pouhé nutnosti porovnat procesní časy operací. V praxi se používá zejména v případech, kdy je nutné vykázat na pracovišti co největší výkon v počtu odvedených operací v daném časovém intervalu.
- LPT (Longest Processing Time) – zakázka s nejdelším procesním časem. Jeho aplikace je jednoduchá stejně jako u SPT. Jeho použití v praxi je v případech, kdy nejprve rozvrhujeme operace s nejdelším procesním časem tak, aby při vzniku tzv. vlomů (nenadálých požadavcích na výrobní zdroje) byla tzv. časová okna pro vložení operací s krátkou dobou trvání co největší, a tak nehrozilo riziko opoždění již rozvržených dlouhotrvajících operací.
- MWKR (Most WorK Remaining) – zakázka, na které zbývá nejvíce práce. Toto prioritní pravidlo patří mezi obtížně aplikovatelné vzhledem k tomu, že je nutné neustále sledovat průběh jednotlivých zakázek s ohledem na předpokládaný čas dokončení operací, které ještě nebyly rozvrženy. Toto pravidlo se používá zejména v předmontážních střediscích, kdy je kladen důraz na co nejmenší časový úsek mezi dokončením operací na zakázkách (dílech), které jsou nutné na následující montážní operace.
- FCFS (First Come First Served) resp. First-In First-Out (FIFO) zakázka s nejdřívějším časem příchodu na pracoviště. Toto pravidlo je v současné době nejpoužívanějším vzhledem k vlivu metod štíhlé výroby a vlivem jeho jednoduché aplikace bez výraznější pomoci od rozvrhovacího nástroje. Zakázky se

jednoduchým způsobem řadí před pracoviště do výrobních front podle času, kdy byly dostupné pro následující zdroj (stroj) zpracovávající zakázku.

- RS (Random Select) - náhodný výběr zakázky. V případech, kdy není jasná priorita zakázky či další požadavky na výrobu (např. montáž) je v praxi často používaným. Toto pravidlo zahrnuje i případy, kdy je sekvence zakázek předem daná, avšak na mistra řídícího výrobu je vyvíjen nátlak, aby priority daných zakázek měnil. V takových případech se mistr rozhoduje náhodně. Toto pravidlo bylo vybráno zvláště pro porovnání přístupů, kdy je rozvrhování řízeno předem danými pravidly a kdy se mistr či dílenský plánovač rozhoduje náhodně vzhledem k momentální situaci.

Popisované algoritmy pro aktivní rozvrh (A) a rozvrh bez zpoždění (ND) s použitím výše zmíněných prioritních pravidel (SPT, LPT, MWKR, FIFO a RS) byly testovány na 10-ti známých rozvrhovacích problémech pro JSSP (první číslo značí počet strojů, druhé pak počet úloh) a 10-ti problémech pro FJSSP určených pro testování heuristických algoritmů pomocí účelové funkce makespan:

- JSSP
 - Fisher H., Thompson G.L. [149]: 6x6 (ft6); 10x10 (ft10); 20x5 (ft20).
 - Storer R.H., Wu S.D., Vaccari R. [150]: 50x10 (sw11).
 - Lawrence S. [151]: 10x5 (La2); 10x10 (La19); 15x10 (La21); 20x10 (La27); 20x10 (La30); 15x15 (La40).
- FJSSP
 - Brandimarte P. [152]: MK01-MK10

Při testování prioritních pravidel se předpokládá, že tato nebudou dosahovat optimálních výsledků účelové funkce makespan a to i s ohledem na velikost prostoru možných řešení (tab. 6-1 a 6-2)

Tab. 6-1: Množina všech řešení testovaných modelů JSSP

Model	Počet zakázek n	Počet strojů m	Počet operací nm	Množina všech řešení $(n!)^m$
FT06	6	6	36	1,39E+17
FT10	10	10	100	3,96E+65
FT20	20	5	100	8,52E+91
SW11	50	10	500	6,77E+644
LA2	10	5	50	6,29E+32
LA19	10	10	100	3,96E+65
LA21	15	10	150	1,46E+121
LA27	20	10	200	7,27E+183
LA30	20	10	200	7,27E+183
LA40	15	15	225	5,59E+181

Tab. 6-2: Odhad velikosti množiny všech řešení testovaných modelů FJSSP

Model	Počet zakázek n	Počet strojů m	Počet operací nm	Průměrný počet variant strojů f	Množina všech řešení $(n!)^{mf}$
MK01	10	6	60	2,09	1,81731E+82
MK02	10	6	60	4,1	2,3451E+161
MK03	15	8	120	3,01	5,8252E+291
MK04	15	8	120	1,91	1,3807E+185
MK05	15	4	60	1,71	7,53108E+82
MK06	10	10	100	3,27	3,1934E+214
MK07	20	5	100	2,83	1,4577E+260
MK08	20	10	200	1,43	8,348E+262
MK09	20	10	200	2,53	1,47E+465
MK10	20	15	300	2,98	7,24E+821

Výsledky testování konstruktivních algoritmů pomocí prioritních pravidel dle účelové funkce makespan jsou uvedeny v tabulce 6-3 (aktivní rozvrhy) a v tabulce 6-4 (rozvrhy bez zpoždění) u rozvrhů JSSP.

V rámci pravidla RS je generováno 10 náhodných rozvrhů, u kterých je sledována nejlepší (RS_b), nejhorší (RS_w) a průměrná (RS_a) hodnota účelové funkce.

Tab. 6-3: Výsledky prioritních pravidel – makespan (aktivní rozvrh modely JSSP)

Model	Optimum	SPT	LPT	MWKR	FIFO	RS_b	RS_w	RS_a
FT06	55	94	86	67	65	65	77	67,4
FT10	930	1429	1355	1178	1184	1352	1456	1386,6
FT20	1165	1675	1661	1556	1645	1539	1757	1627,6
SW11	2983	4682	4678	4492	4427	4045	4414	4266,9
LA2	655	1055	1025	860	830	912	1089	982,9
LA19	842	1286	1120	1084	983	1167	1248	1197,3
LA21	1046	1657	1560	1267	1265	1559	1731	1648
LA27	1235	2201	1737	1567	1644	1890	2072	1973
LA30	1355	2194	1871	1565	1648	1990	2320	2134,9
LA40	1222	1804	1914	1549	1531	1820	2239	2043,2

Tab. 6-4: Výsledky prioritních pravidel – makespan (rozvrh bez zpoždění modely JSSP)

Model	Optimum	SPT	LPT	MWKR	FIFO	RS_b	RS_w	RS_a
FT06	55	88	77	61	65	66	73	68,4
FT10	930	1074	1295	1108	1184	1155	1394	1258,2
FT20	1165	1267	1631	1501	1645	1385	1518	1476,7
SW11	2983	3530	4763	4342	4427	3917	4199	4068,3
LA2	655	756	940	813	830	793	871	838,5
LA19	842	940	1062	1013	983	1039	1052	1050,1
LA21	1046	1324	1451	1264	1265	1292	1516	1397,4
LA27	1235	1784	1644	1442	1644	1610	1849	1724
LA30	1355	1713	1902	1534	1648	1637	1889	1785,7
LA40	1222	1476	1822	1440	1531	1434	1790	1562,4

Z výsledků je patrné (Tab. 6-3 a 6-4 - žlutá barva naznačuje nejlepší výsledek ze všech, zelená pak u daného typu rozvrhu A nebo ND), že u aktivních rozvrhů dosahují pravidla FIFO a MWKR nejlepších hodnot. Překvapivě dobrých výsledků dosahuje také pravidlo RS, které má dva nejlepší výsledky v rámci aktivních rozvrhů a v průměru má lepší výsledky než hojně používané SPT.

Celkově nejlepšího počtu výsledků však dosahují rozvrhy generované pomocí rozvrhů bez zpoždění a to zejména pravidlo SPT a MWKR.

Výsledky testovaných flexibilních modelů (FJSSP) jsou zobrazeny v tabulce 6-5 (aktivní rozvrhy) a tabulce 6-6 (rozvrhy bez zpoždění).

Tab. 6-5: Výsledky prioritních pravidel – makespan (aktivní rozvrh modely FJSSP)

Model	Optimum	SPT	LPT	MWKR	FIFO	RS_b	RS_w	RS_a
MK01	36	81	61	51	45	56	73	66,7
MK02	24	75	68	39	38	50	70	58,5
MK03	204	458	305	219	204	279	325	301
MK04	48	103	153	80	81	96	159	128,6
MK05	168	314	232	198	199	234	267	250,4
MK06	33	115	139	94	106	112	134	121,8
MK07	133	231	293	208	208	232	267	248,1
MK08	523	768	673	524	531	652	754	703,9
MK09	299	650	521	369	350	483	546	513,8
MK10	165	510	513	287	291	404	474	435,6

Tab. 6-6: Výsledky prioritních pravidel – makespan (rozvrh bez zpoždění modely FJSSP)

Model	Optimum	SPT	LPT	MWKR	FIFO	RS_b	RS_w	RS_a
MK01	36	51	60	51	47	51	62	58,6
MK02	24	65	50	44	36	51	76	59,5
MK03	204	301	250	204	208	254	322	278,4
MK04	48	78	103	77	82	80	98	90
MK05	168	247	211	200	189	205	239	222,4
MK06	33	122	116	105	93	93	131	113,1
MK07	133	237	295	200	213	254	289	254
MK08	523	651	695	523	531	611	647	629,3
MK09	299	489	441	349	365	390	502	439,1
MK10	165	354	398	266	287	343	375	357,3

Výsledky ukazují (tab. 6-5 a 6-6), že při testování modelů FJSSP dosahují při použití aktivních rozvrhů i rozvrhů bez zpoždění nejlepších výsledků pravidla FIFO a MWKR. Celkově nejlepších výsledků (až na model MK01) pak bylo dosaženo pomocí rozvrhů bez zpoždění.

Druhým důležitým hlediskem při testování konstruktivního algoritmu a prioritních pravidel je pak odchylka od známého optima (JSSP tab. 6-7 a 6-8, pro FJSSP tab. 6-9 a 6-10).

Tab. 6-7: Odchylka od optima prioritních pravidel – makespan (aktivní rozvrh modely JSSP)

Model	Optimum	SPT	LPT	MWKR	FIFO	RS_b	RS_w	RS_a
FT06	55	70,91%	56,36%	21,82%	18,18%	18,18%	40,00%	22,55%
FT10	930	53,66%	45,70%	26,67%	27,31%	45,38%	56,56%	49,10%
FT20	1165	43,78%	42,58%	33,56%	41,20%	32,10%	50,82%	39,71%
SW11	2983	56,96%	56,82%	50,59%	48,41%	35,60%	47,97%	43,04%
LA2	655	61,07%	56,49%	31,30%	26,72%	39,24%	66,26%	50,06%
LA19	842	52,73%	33,02%	28,74%	16,75%	38,60%	48,22%	42,20%
LA21	1046	58,41%	49,14%	21,13%	20,94%	49,04%	65,49%	57,55%
LA27	1235	78,22%	40,65%	26,88%	33,12%	53,04%	67,77%	59,76%
LA30	1355	61,92%	38,08%	15,50%	21,62%	46,86%	71,22%	57,56%
LA40	1222	47,63%	56,63%	26,76%	25,29%	48,94%	83,22%	67,20%
	Průměr	58,53%	47,55%	28,29%	27,95%	40,70%	59,75%	48,87%

Při použití aktivních (tab. 6-7) rozvrhů dosahuje odchylka od optima průměrně 27,95% - 59,75%, přičemž v případě FIFO rozvrhů, které jsou u aktivních rozvrhů nejúspěšnější, ve více než u poloviny případů pod 30%.

Tab. 6-8: Odchylka od optima prioritních pravidel – makespan (rozvrh bez zpoždění modely JSSP)

Model	Optimum	SPT	LPT	MWKR	FIFO	RS_b	RS_w	RS_a
FT06	55	60,00%	40,00%	10,91%	18,18%	20,00%	32,73%	24,36%
FT10	930	15,48%	39,25%	19,14%	27,31%	24,19%	49,89%	35,29%
FT20	1165	8,76%	40,00%	28,84%	41,20%	18,88%	30,30%	26,76%
SW11	2983	18,34%	59,67%	45,56%	48,41%	31,31%	40,76%	36,38%
LA2	655	15,42%	43,51%	24,12%	26,72%	21,07%	32,98%	28,02%
LA19	842	11,64%	26,13%	20,31%	16,75%	23,40%	24,94%	24,71%
LA21	1046	26,58%	38,72%	20,84%	20,94%	23,52%	44,93%	33,59%
LA27	1235	44,45%	33,12%	16,76%	33,12%	30,36%	49,72%	39,60%
LA30	1355	26,42%	40,37%	13,21%	21,62%	20,81%	39,41%	31,79%
LA40	1222	20,79%	49,10%	17,84%	25,29%	17,35%	46,48%	27,86%
Průměr		24,79%	40,99%	21,75%	27,95%	23,09%	39,21%	30,84%

Rozvrhy bez zpoždění (tab. 6-8) v průměru dosahují menších odchylek od optima než rozvrhy aktivní, přičemž jako nejúspěšnější pravidlo u vybraných modelů lze považovat MWKR, které v průměru dosahovalo 21,75% odchylku od optima a stalo se tak v kombinaci s rozvrhy bez zpoždění nejúspěšnější ze všech variant.

Tab. 6-9: Odchylka od optima prioritních pravidel – makespan (aktivní rozvrh modely FJSSP)

Model	Optimum	SPT	LPT	MWKR	FIFO	RS_b	RS_w	RS_a
MK01	36	125,00%	69,44%	41,67%	25,00%	55,56%	102,78%	85,28%
MK02	24	212,50%	183,33%	62,50%	58,33%	108,33%	191,67%	143,75%
MK03	204	124,51%	49,51%	7,35%	0,00%	36,76%	59,31%	47,55%
MK04	48	114,58%	218,75%	66,67%	68,75%	100,00%	231,25%	167,92%
MK05	168	86,90%	38,10%	17,86%	18,45%	39,29%	58,93%	49,05%
MK06	33	248,48%	321,21%	184,85%	221,21%	239,39%	306,06%	269,09%
MK07	133	73,68%	120,30%	56,39%	56,39%	74,44%	100,75%	86,54%
MK08	523	46,85%	28,68%	0,19%	1,53%	24,67%	44,17%	34,59%
MK09	299	117,39%	74,25%	23,41%	17,06%	61,54%	82,61%	71,84%
MK10	165	209,09%	210,91%	73,94%	76,36%	144,85%	187,27%	164,00%
Průměr		135,90%	131,45%	53,48%	54,31%	88,48%	136,48%	111,96%

U aktivních rozvrhů modelů (tab. 6-9) FJSSP dosahuje odchylka od optima průměrně přibližně 54-135%. Rozdíly mezi jednotlivými odchylkami jsou značné. Pravidlo MWKR, které v průměru dosahuje 53,45% odchylky od optima a dokonce nalézá řešení velmi blízké optimu (MK08 – 0,19%) má maximální odchylku 184,85%. Druhé nejúspěšnější pravidlo FIFO dosahuje podobných výsledků, v případě MK03 nalézá optimum, avšak jeho maximální odchylka je více než 200% (MK06).

Tab. 6-10: Odchylka od optima prioritních pravidel – makespan (rozvrh bez zpoždění)

Model	Optimum	SPT	LPT	MWKR	FIFO	RS_b	RS_w	RS_a
MK01	36	41,67%	66,67%	41,67%	30,56%	41,67%	72,22%	62,78%
MK02	24	170,83%	108,33%	83,33%	50,00%	112,50%	216,67%	147,92%
MK03	204	47,55%	22,55%	0,00%	1,96%	24,51%	57,84%	36,47%
MK04	48	62,50%	114,58%	60,42%	70,83%	66,67%	104,17%	87,50%
MK05	168	47,02%	25,60%	19,05%	12,50%	22,02%	42,26%	32,38%
MK06	33	269,70%	251,52%	218,18%	181,82%	181,82%	296,97%	242,73%
MK07	133	78,20%	121,80%	50,38%	60,15%	90,98%	117,29%	90,98%
MK08	523	24,47%	32,89%	0,00%	1,53%	16,83%	23,71%	20,33%
MK09	299	63,55%	47,49%	16,72%	22,07%	30,43%	67,89%	46,86%
MK10	165	114,55%	141,21%	61,21%	73,94%	107,88%	127,27%	116,55%
Průměr		92,00%	93,26%	55,10%	50,54%	69,53%	112,63%	88,45%

Rozvrhy bez zpoždění (tab. 6-10) dosahují na testovaných modelech FJSSP v průměru menších odchylek použitím prioritních pravidel než je tomu v případě aktivních rozvrhů. Nejlepším pravidlem je pak pravidlo FIFO, které dosahuje v průměru 50,54% odchylky od optima. Stejně jako v přechozím případě (aktivní rozvrhy) je rozsah odchylek značný 1,53%-181,82% a tudíž jeho použitím nelze v případě testovaných modelů dosáhnout vyrovnané úspěšnosti dosažení optima. Tím se potvrzuje předpoklad, že pro různé případy může být vhodné jiné prioritní pravidlo, tzn. nelze paušalizovat vhodnost použití toho či onoho pravidla.

Z výsledků tohoto testování lze usoudit, že ani jedno z aplikovaných pravidel nelze obecně použít k dosažení optimálních výsledků na jakýkoli problém, přestože FIFO a MWKR dosahovalo v průměru nejlepších výsledků. Použití jednoho prioritního pravidla v jednom modelu výrobního systému nemusí poskytnout kvalitní výsledek v případě modelu jiného. Pro analýzu a optimalizaci rozvrhů pomocí prioritních pravidel se tedy jako vhodné jeví kombinace několika prioritních pravidel.

Posledním sledovaným kritériem testování teoretických modelů je sledování závislosti časové náročnosti tvorby rozvrhu na počtu rozvrhovaných operací (tab. 6-11 a 6-12). V tomto testování byl měřen celkový čas generování všech rozvrhů dle obou alternativ generování A a ND s použitím všech pravidel. Celkový počet rozvrhů je tedy celkem 28 ($2 \times \{SPT, LPT, MWKR\} + 2 \times \{10 RS\}$).

Tab. 6-11: Časová náročnost rozvrhování JSSP

Model	Celkový čas tvorby rozvrhů (s)	Čas tvorby jednoho rozvrhu (ms)	Počet operací nm	Čas na jednu operaci (ms)
FT06	1,2	42,86	36	1,19
LA2	1,25	44,64	50	0,89
FT10	3,2	114,29	100	1,14
FT20	3,3	117,86	100	1,18
LA19	3	107,14	100	1,07
LA21	4,6	164,29	150	1,10
LA27	6,1	217,86	200	1,09
LA30	6	214,29	200	1,07
LA40	7,3	260,71	225	1,16
SW11	20,55	733,93	500	1,47

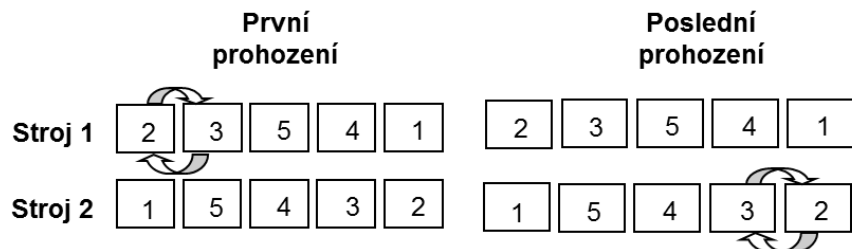
Tab. 6-12: Časová náročnost rozvrhování

Model	Celkový čas tvorby rozvrhů (s)	Čas tvorby jednoho rozvrhu (ms)	Počet operací nm	Čas na jednu operaci (ms)
MK01	1,8	64,29	60	1,07
MK02	2,1	75,00	60	1,25
MK03	5,9	210,71	120	1,76
MK04	3	107,14	120	0,89
MK05	3,1	110,71	60	1,85
MK06	5,5	196,43	100	1,96
MK07	3,7	132,14	100	1,32
MK08	7,8	278,57	200	1,39
MK09	9,4	335,71	200	1,68
MK10	11,3	403,57	300	1,35

Jak je patrné z naměřených dat, není závislost počtu operací na celkovém čase tvorby rozvrhu lineární. To je způsobeno nestejnou dobou trvání jedné iterace rozvrhovacího algoritmu, která je závislá na velikosti konfliktní množiny operací, ze kterých se vybírá jedna dle prioritního pravidla.

6.2 Algoritmus lokálního prohledávání

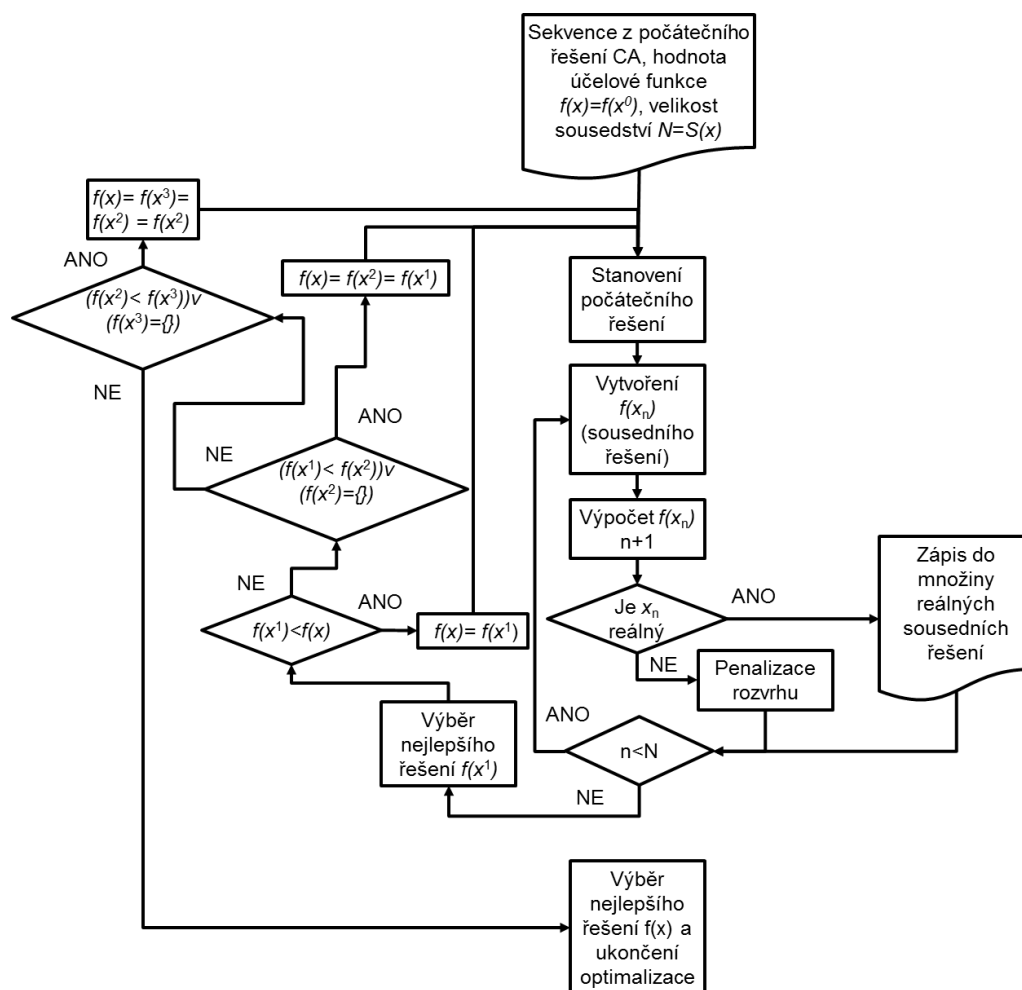
Pro testování lokálního prohledávání je vytvořen jednoduchý algoritmus lokálního prohledávání (LS) jedním prohozením. Tento typ prohledávání nepoužívá žádnou znalost o problému. Prohledávání okolí je definované jako jednotlivé prohazování pořadí operací na každém stroji viz obr. 6-1 (číslo v bloku je číslem zakázky).



Obr. 6-1: Procedura prohledávání okolí jedním prohozením

Samotný algoritmus nejprve načte sekvenci vygenerovanou jako počáteční řešení. Pomocí výše popsané procedury vygeneruje sousední řešení. Procedura generuje (prohledává) většinou nepřijatelná řešení, tj. generuje řešení, která nesplňují pravidla aktivního rozvrhu a rozvrhu bez zpoždění. V případě, že sousední řešení je přijatelné, zaznamená se hodnota účelové funkce a po skončení ohodnocení všech vygenerovaných řešení se vybere nejlepší řešení.

Pokud je toto řešení lepší než počáteční, je toto řešení použito jako nově výchozí. V opačném případě je porovnáno na základě hodnoty účelové funkce s předchozími řešeními. Současné řešení není přijato pouze tehdy, když v předchozích dvou iteracích nedošlo k zlepšení počátečního řešení. Tehdy je optimalizace ukončena a poskytne nejlepší z vygenerovaných rozvrhů (obráz. 6-2 diagram pro minimalizační problémy).



Obr. 6-2: Diagram algoritmu založeného na lokálním prohledávání

Pro testování prohledávacího algoritmu jsou použity jak modely JSSP, tak modely FJSSP. Jako počáteční řešení jsou použity rozvrhy vygenerované pomocí prioritních pravidel z předchozí kapitoly a to včetně pravidel vygenerovaných pomocí RS, které bývají nejčastěji používány pro generování počátečních řešení.

Při testování tohoto algoritmu bylo sledováno několik ukazatelů. Prvním z nich je hodnota účelové funkce makespan (tab. 6-13 a 6-14 pro JSSP, tab. 6-15 a 6-16 pro FJSSP). Dále je sledováno zlepšení účelové funkce oproti počátečním řešením (tab. 6-17 a 6-18 pro JSSP a tab. 6-19 a 6-20 pro FJSSP). Důležitým kritériem využití tohoto algoritmu je také schopnost metody dosáhnout optimálních řešení (tab. 6-21 pro JSSP a 6-22 pro FJSSP).

Tab. 6-13: Hodnoty účelové funkce makespan použitím LS – aktivní rozvrhy JSSP

Model	Optimum	SPT	LS	LPT	LS	MWKR	LS	FIFO	LS	RS_b	LS_b
FT06	55	94	94	86	71	67	58	65	55	65	64
FT10	930	1429	1293	1355	1283	1178	1052	1184	1172	1352	1307
FT20	1165	1675	1596	1661	1593	1556	1501	1645	1598	1539	1460
SW11	2983	4682	4585	4678	4593	4492	4433	4427	4348	4045	3924
LA2	655	1055	834	1025	895	860	829	830	733	912	762
LA19	842	1286	1182	1120	1062	1084	1029	983	966	1167	1160
LA21	1046	1657	1581	1560	1426	1267	1213	1265	1209	1559	1497
LA27	1235	2201	2146	1737	1702	1567	1545	1644	1582	1890	1657
LA30	1355	2194	2140	1871	1806	1565	1523	1648	1560	1990	1869
LA40	1222	1804	1681	1914	1740	1549	1539	1531	1380	1820	1810

Tab. 6-14: Hodnoty účelové funkce makespan použitím LS –rozvrhy bez zpoždění JSSP

Model	Optimum	SPT	LS	LPT	LS	MWKR	LS	FIFO	LS	RS_b	LS_a
FT06	55	88	75	77	74	61	60	65	60	66	66
FT10	930	1074	1074	1295	1241	1108	1108	1184	1184	1155	1155
FT20	1165	1267	1267	1631	1618	1501	1501	1645	1621	1385	1320
SW11	2983	3530	3530	4763	4763	4342	4342	4427	4427	3917	3874
LA2	655	756	756	940	940	813	813	830	830	793	793
LA19	842	940	940	1062	1062	1013	1008	983	966	1039	979
LA21	1046	1324	1324	1451	1451	1264	1264	1265	1265	1292	1274
LA27	1235	1784	1784	1644	1610	1442	1442	1644	1612	1610	1578
LA30	1355	1713	1713	1902	1902	1534	1534	1648	1648	1637	1608
LA40	1222	1476	1476	1822	1780	1440	1413	1531	1394	1434	1402

Optimalizací aktivních rozvrhů pomocí metody lokálního prohledávání jsou nalezeny lepší výsledky, než tomu je u rozvrhů bez zpoždění. V 6-ti z 10-ti (tab. 6-13 žlutě) případů poskytlo LS s A lepší výsledky než LS s ND. Jako počáteční řešení jsou u těchto modelů nejvhodnější prioritní pravidla MWKR a FIFO. Počáteční řešení generovaná pomocí RS se ukázala navzdory očekávání u většiny případů jako nevhodná.

Lokální prohledávání pomocí rozvrhů bez zpoždění dosáhlo u JSSP ze všech generovaných rozvrhů nejlepších řešení 4 z 10-ti (tab. 6-14 žlutě), avšak u 3 z těchto výsledků se jedná o počáteční řešení generované prioritními pravidly.

Tab. 6-15: Hodnoty účelové funkce makespan použitím LS – aktivní rozvrhy FJSSP

Model	Optimum	SPT	LS	LPT	LS	MWKR	LS	FIFO	LS	RS_b	LS_a
MK01	36	81	77	61	61	51	51	45	45	56	50
MK02	24	75	75	68	68	39	39	38	38	50	46
MK03	204	458	416	305	305	219	217	204	204	279	279
MK04	48	103	100	153	153	80	80	81	79	96	91
MK05	168	314	313	232	229	198	198	199	199	234	234
MK06	33	115	114	139	139	94	94	106	106	112	112
MK07	133	231	230	293	293	208	208	208	208	232	224
MK08	523	768	725	673	662	524	524	531	523	652	652
MK09	299	650	650	521	521	369	369	350	350	483	479
MK10	165	510	510	513	513	287	287	291	291	404	404

Tab. 6-16: Hodnoty účelové funkce makespan použitím LS –rozvrhy bez zpoždění FJSSP

Model	Optimum	SPT	LS	LPT	LS	MWKR	LS	FIFO	LS	RS_b	LS_a
MK01	36	51	51	60	60	51	51	47	46	51	51
MK02	24	65	65	50	50	44	44	36	36	51	46
MK03	204	301	301	250	250	204	204	208	206	254	251
MK04	48	78	78	103	103	77	77	82	80	80	80
MK05	168	247	247	211	211	200	200	189	189	205	205
MK06	33	122	122	116	116	105	105	93	93	93	93
MK07	133	237	237	295	295	200	200	213	213	254	254
MK08	523	651	651	695	695	523	523	531	523	611	604
MK09	299	489	489	441	441	349	349	365	365	390	390
MK10	165	354	354	398	398	266	266	287	287	343	343

U modelů FJSSP je patrné, že navržené schéma LS optimalizace manipuluje se sekvencí před danými pracovišti a ne s alokací operací na záměnná pracoviště tak jak to FJSSP vyžaduje. Z tohoto důvodu vykazuje LS zlepšení účelové funkce jen výjimečně.

U většiny aktivních rozvrhů vyjma modelu MK04 a MK08, u kterého došlo k nalezení optima v případě aktivních rozvrhů, nezlepšilo LS své počáteční řešení.

Většina nalezených nejlepších výsledků lokálním prohledáváním je pomocí rozvrhu bez zpoždění (tab. 6-16 žlutě). Je však nutné brát ohled na zlepšení vůči počátečnímu řešení. Pouze u modelu MK01 našlo LS lepší řešení než je to počáteční (generované pomocí CA a FIFO).

Pro další posouzení přínosu navrženého algoritmu LS pro modely JSSP (tab. 6-17 a 6-18) a FJSSP (tab. 6-19 a 6-20) je nutné analyzovat procentuální zlepšení výsledků vůči CA.

Tab. 6-17: Zlepšení účelové funkce makespan použitím LS – aktivní rozvrhy JSSP

Model	Optimum	Zlepšení				
		SPT/LS	LPT/LS	MKWR/LS	FIFO/LS	RS/LS_b
FT06	55	0,00%	27,27%	16,36%	18,18%	1,82%
FT10	930	14,62%	7,74%	13,55%	1,29%	4,84%
FT20	1165	6,78%	5,84%	4,72%	4,03%	6,78%
SW11	2983	3,25%	2,85%	1,98%	2,65%	4,06%
LA2	655	33,74%	19,85%	4,73%	14,81%	22,90%
LA19	842	12,35%	6,89%	6,53%	2,02%	0,83%
LA21	1046	7,27%	12,81%	5,16%	5,35%	5,93%
LA27	1235	4,45%	2,83%	1,78%	5,02%	18,87%
LA30	1355	3,99%	4,80%	3,10%	6,49%	8,93%
LA40	1222	10,07%	14,24%	0,82%	12,36%	0,82%
Průměrné zlepšení		9,65%	10,51%	5,87%	7,22%	7,58%

Jak je patrné z tabulky 6-17, LS zlepšuje počáteční řešení průměrně o 5,87%-10,51%, přičemž nejlepších výsledků z hlediska zlepšení je dosaženo pomocí rozvrhů vycházející z LPT. Z posouzení jednotlivých zlepšení je patrné, že ačkoli lze pomocí LS dosáhnout radikálních zlepšení (např. SPT/LS u LA2 33,74%), v některých případech LS není schopné počáteční řešení zlepšit, tzn. LS uvízne v lokálním extrému (např. FT06 SPT/LS)

Tab. 6-18: Zlepšení účelové funkce makespan použitím LS –rozvrhy bez zpoždění JSSP

Model	Optimum	Zlepšení				
		SPT/LS	LPT/LS	MKWR/LS	FIFO/LS	RS/LS_a
FT06	55	14,77%	3,90%	1,64%	7,69%	1,92%
FT10	930	0,00%	4,17%	0,00%	0,00%	0,25%
FT20	1165	0,00%	0,80%	0,00%	1,46%	2,22%
SW11	2983	0,00%	0,00%	0,00%	0,00%	1,04%
LA2	655	0,00%	0,00%	0,00%	0,00%	1,56%
LA19	842	0,00%	0,00%	0,49%	1,73%	1,58%
LA21	1046	0,00%	0,00%	0,00%	0,00%	0,87%
LA27	1235	0,00%	2,07%	0,00%	1,95%	1,66%
LA30	1355	0,00%	0,00%	0,00%	0,00%	2,36%
LA40	1222	0,00%	2,31%	1,88%	8,95%	1,37%
Průměrné zlepšení		1,48%	1,32%	0,40%	2,18%	1,48%

Rozvrhy optimalizované pomocí LS v oblasti rozvrhů bez zpoždění dosahují jen výjimečně zlepšení (tab. 6-18). Průměrné zlepšení dosahuje pouze 0,4%-2,18% při maximálním zlepšení 14,77%. Důvodem je jako u aktivních rozvrhů uvíznutí v lokálním extrému.

Tab. 6-19: Zlepšení účelové funkce makespan použitím LS – aktivní rozvrhy FJSSP

Model	Optimum	Zlepšení				
		SPT/LS	LPT/LS	MKWR/LS	FIFO/LS	RS/LS_b
MK01	36	11,11%	0,00%	0,00%	0,00%	16,67%
MK02	24	0,00%	0,00%	0,00%	0,00%	16,67%
MK03	204	20,59%	0,00%	0,98%	0,00%	0,00%
MK04	48	6,25%	0,00%	0,00%	4,17%	10,42%
MK05	168	0,60%	1,79%	0,00%	0,00%	0,00%
MK06	33	3,03%	0,00%	0,00%	0,00%	0,00%
MK07	133	0,75%	0,00%	0,00%	0,00%	6,02%
MK08	523	8,22%	2,10%	0,00%	1,53%	0,00%
MK09	299	0,00%	0,00%	0,00%	0,00%	1,34%
MK10	165	0,00%	0,00%	0,00%	0,00%	0,00%
Průměrné zlepšení		5,05%	0,39%	0,10%	0,57%	5,11%

Přes to, že navržený způsob optimalizace nemanipuluje s přiřazováním operací na stroje, je navržený algoritmus LS schopný v některých případech zlepšit počáteční řešení (tab. 6-19). Rozsah zlepšení se pohybuje mezi 0,1%-5,11%. Ve většině případů však ke zlepšení nedochází. To je způsobeno jednak výše popsaným mechanismem LS a zároveň počátečními řešeními, která jsou natolik silnými lokálními extrémy, že LS je není schopné dále optimalizovat.

Tab. 6-20: Zlepšení účelové funkce makespan použitím LS –rozvrhy bez zpoždění FJSSP

Model	Optimum	Zlepšení				
		SPT/LS	LPT/LS	MKWR/LS	FIFO/LS	RS/LS_b
MK01	36	0,00%	0,00%	0,00%	2,78%	0,00%
MK02	24	0,00%	0,00%	0,00%	0,00%	20,83%
MK03	204	0,00%	0,00%	0,00%	0,98%	1,47%
MK04	48	0,00%	0,00%	0,00%	4,17%	0,00%
MK05	168	0,00%	0,00%	0,00%	0,00%	0,00%
MK06	33	0,00%	0,00%	0,00%	0,00%	0,00%
MK07	133	0,00%	0,00%	0,00%	0,00%	0,00%
MK08	523	0,00%	0,00%	0,00%	1,53%	1,34%
MK09	299	0,00%	0,00%	0,00%	0,00%	0,00%
MK10	165	0,00%	0,00%	0,00%	0,00%	0,00%
Průměrné zlepšení		0,00%	0,00%	0,00%	0,95%	2,36%

V případě rozvrhů bez zpoždění u modelů FJSSP nedochází až na výjimky k zlepšení. Těmito výjimkami je aplikování LS na počáteční řešení z pravidel FIFO a LS, ale to pouze v 7-mi případech z 20-ti, kde zlepšení se pohybuje mezi 0,98%-20,83%. Stejně jako v předešlých případech je to způsobeno silnými lokálními extrémy počátečních řešení a navrženým mechanismem optimalizace LS.

Jedním z hlavních parametrů algoritmu je pak schopnost dosáhnout optima nebo optimu blízkým výsledkům účelové funkce (tab. 6-21 a 6-22 modely JSSP, tab. 6-23 a 6-24 modely FJSSP).

Tab. 6-21: Vzdálenost nalezeného makespan od optima – aktivní rozvrhy JSSP

Model	Optimum	Odchyka od optima									
		SPT	SPT/LS	LPT	LPT/LS	MKWR	MKWR/LS	FIFO	FIFO/LS	RS_b	RS/LS_b
FT06	55	70,91%	70,91%	56,36%	29,09%	21,82%	5,45%	18,18%	0,00%	18,18%	16,36%
FT10	930	53,66%	39,03%	45,70%	37,96%	26,67%	13,12%	27,31%	26,02%	45,38%	40,54%
FT20	1165	43,78%	37,00%	42,58%	36,74%	33,56%	28,84%	41,20%	37,17%	32,10%	25,32%
SW11	2983	56,96%	53,70%	56,82%	53,97%	50,59%	48,61%	48,41%	45,76%	35,60%	31,55%
LA2	655	61,07%	27,33%	56,49%	36,64%	31,30%	26,56%	26,72%	11,91%	39,24%	16,34%
LA19	842	52,73%	40,38%	33,02%	26,13%	28,74%	22,21%	16,75%	14,73%	38,60%	37,77%
LA21	1046	58,41%	51,15%	49,14%	36,33%	21,13%	15,97%	20,94%	15,58%	49,04%	43,12%
LA27	1235	78,22%	73,77%	40,65%	37,81%	26,88%	25,10%	33,12%	28,10%	53,04%	34,17%
LA30	1355	61,92%	57,93%	38,08%	33,28%	15,50%	12,40%	21,62%	15,13%	46,86%	37,93%
LA40	1222	47,63%	37,56%	56,63%	42,39%	26,76%	25,94%	25,29%	12,93%	48,94%	48,12%
Průměrná odchyka		58,53%	48,88%	47,55%	37,03%	28,29%	22,42%	27,95%	20,73%	40,70%	33,12%

Tab. 6-22: Vzdálenost nalezeného makespan od optima –rozvrhy bez zpoždění JSSP

Model	Optimum	Odchyka od optima									
		SPT	SPT/LS	LPT	LPT/LS	MKWR	MKWR/LS	FIFO	FIFO/LS	RS_b	RS/LS_b
FT06	55	60,00%	36,36%	40,00%	34,55%	10,91%	9,09%	18,18%	9,09%	20,00%	20,00%
FT10	930	15,48%	15,48%	39,25%	33,44%	19,14%	19,14%	27,31%	27,31%	24,19%	24,19%
FT20	1165	8,76%	8,76%	40,00%	38,88%	28,84%	28,84%	41,20%	39,14%	18,88%	13,30%
SW11	2983	18,34%	18,34%	59,67%	59,67%	45,56%	45,56%	48,41%	48,41%	31,31%	29,87%
LA2	655	15,42%	15,42%	43,51%	43,51%	24,12%	24,12%	26,72%	26,72%	21,07%	21,07%
LA19	842	11,64%	11,64%	26,13%	26,13%	20,31%	19,71%	16,75%	14,73%	23,40%	16,27%
LA21	1046	26,58%	26,58%	38,72%	38,72%	20,84%	20,84%	20,94%	20,94%	23,52%	21,80%
LA27	1235	44,45%	44,45%	33,12%	30,36%	16,76%	16,76%	33,12%	30,53%	30,36%	27,77%
LA30	1355	26,42%	26,42%	40,37%	40,37%	13,21%	13,21%	21,62%	21,62%	20,81%	18,67%
LA40	1222	20,79%	20,79%	49,10%	45,66%	17,84%	15,63%	25,29%	14,08%	17,35%	14,73%
Průměrná odchyka		24,79%	22,42%	40,99%	39,13%	21,75%	21,29%	27,95%	25,26%	23,09%	20,77%

Z tabulek 6-21 a 6-22 je patrné, že ačkoli navržený LS je jednoduchou metodou, dokáže nalézt bližší výsledky výsledkům optimálním než CA. Rozdíl mezi vzdálenostmi od optima je cca 7%-10% u aktivních rozvrhů, u rozvrhů bez zpoždění pouze cca 0,5%-2,5%. Největšího průměrného přiblížení pak je dosaženo aktivními rozvrhy s počátečním řešením generovaným pomocí prioritního pravidla FIFO a pomocí náhodného pravidla u rozvrhů bez zpoždění. U tohoto pravidla je si však nutné uvědomit, že reprezentuje nejlepší řešení z 10-ti možných.

Tab. 6-23: Vzdálenost nalezeného makespan od optima – aktivní rozvrhy FJSSP

Model	Optimum	Odchyka od optima									
		SPT	SPT/LS	LPT	LPT/LS	MKWR	MKWR/LS	FIFO	FIFO/LS	RS_b	RS/LS_b
MK01	36	125,00%	113,89%	69,44%	69,44%	41,67%	41,67%	25,00%	25,00%	55,56%	38,89%
MK02	24	212,50%	212,50%	183,33%	183,33%	62,50%	62,50%	58,33%	58,33%	108,33%	91,67%
MK03	204	124,51%	103,92%	49,51%	49,51%	7,35%	6,37%	0,00%	0,00%	36,76%	36,76%
MK04	48	114,58%	108,33%	218,75%	218,75%	66,67%	66,67%	68,75%	64,58%	100,00%	89,58%
MK05	168	86,90%	86,31%	38,10%	36,31%	17,86%	17,86%	18,45%	18,45%	39,29%	39,29%
MK06	33	248,48%	245,45%	321,21%	321,21%	184,85%	184,85%	221,21%	221,21%	239,39%	239,39%
MK07	133	73,68%	72,93%	120,30%	120,30%	56,39%	56,39%	56,39%	56,39%	74,44%	68,42%
MK08	523	46,85%	38,62%	28,68%	26,58%	0,19%	0,19%	1,53%	0,00%	24,67%	24,67%
MK09	299	117,39%	117,39%	74,25%	74,25%	23,41%	23,41%	17,06%	17,06%	61,54%	60,20%
MK10	165	209,09%	209,09%	210,91%	210,91%	73,94%	73,94%	76,36%	76,36%	144,85%	144,85%
Průměrná odchyka		135,90%	130,84%	131,45%	131,06%	53,48%	53,38%	54,31%	53,74%	88,48%	83,37%

Tab. 6-24: Vzdálenost nalezeného makespan od optima –rozvrhy bez zpoždění FJSSP

Model	Optimum	Odchyka od optima									
		SPT	SPT/LS	LPT	LPT/LS	MKWR	MKWR/LS	FIFO	FIFO/LS	RS b	RS/LS b
MK01	36	41,67%	41,67%	66,67%	66,67%	41,67%	41,67%	30,56%	27,78%	41,67%	41,67%
MK02	24	170,83%	170,83%	108,33%	108,33%	83,33%	83,33%	50,00%	50,00%	112,50%	91,67%
MK03	204	47,55%	47,55%	22,55%	22,55%	0,00%	0,00%	1,96%	0,98%	24,51%	23,04%
MK04	48	62,50%	62,50%	114,58%	114,58%	60,42%	60,42%	70,83%	66,67%	66,67%	66,67%
MK05	168	47,02%	47,02%	25,60%	25,60%	19,05%	19,05%	12,50%	12,50%	22,02%	22,02%
MK06	33	269,70%	269,70%	251,52%	251,52%	218,18%	218,18%	181,82%	181,82%	181,82%	181,82%
MK07	133	78,20%	78,20%	121,80%	121,80%	50,38%	50,38%	60,15%	60,15%	90,98%	90,98%
MK08	523	24,47%	24,47%	32,89%	32,89%	0,00%	0,00%	1,53%	0,00%	16,83%	15,49%
MK09	299	63,55%	63,55%	47,49%	47,49%	16,72%	16,72%	22,07%	22,07%	30,43%	30,43%
MK10	165	114,55%	114,55%	141,21%	141,21%	61,21%	61,21%	73,94%	73,94%	107,88%	107,88%
Průměrná odchylka		92,00%	92,00%	93,26%	93,26%	55,10%	55,10%	50,54%	49,59%	69,53%	67,17%

U modelů FJSSP (tab. 6-23 a 6-24), jak již je zřejmé z analýzy zlepšení počátečních řešení (tab. 6-21 a 6-22), má použití navrhnutého LS z pohledu dosažení lepších hodnot účelové funkce malý přínos. Ve většině případů se dané řešení přiblížilo k optimu pouze o méně než 1%.

Hlavním kritériem při posouzení aplikace takto navrženého algoritmu je jeho přínos založený na posouzení zkrácení celkového času dokončení všech zakázek (makespan) a času potřebného pro optimalizaci. Vzhledem k tomu, že u teoretických modelů jsou udávány časové jednotky bez řádů jednotek, je při této analýze uvažování zlepšení v řádech minut (tab. 6-25). Jako základ porovnání slouží nejlepší výsledek dosažený pomocí LS a CA bez rozdílu druhu rozvrhu (aktivní či bez zpoždění). Zlepšení pak udává rozdíl mezi výsledkem metody CA a LS s tím, že celkový přínos je zlepšení ponížené o nutnou časovou investici do metody LS.

Tab. 6-25: Porovnání časového přínosu optimalizace LS - JSSP

Model	Nejlepší řešení CA [hod:min:sek]	Čas CA [hod:min:sek]	Nejlepší řešení LS [hod:min:sek]	Čas LS [hod:min:sek]	Zlepšení [hod:min:sek]	Přínos [hod:min:sek]
FT06	1:01:00	0:00:01	0:55:00	0:00:03	0:06:00	0:05:57
FT10	17:54:00	0:00:01	17:32:00	0:00:31	0:22:00	00:21:29
FT20	21:07:00	0:00:03	21:07:00	0:01:08	0:00:00	- 0:01:08
SW11	58:50:00	0:00:03	58:50:00	2:15:26	0:00:00	- 2:15:26
LA2	12:36:00	0:00:03	12:13:00	0:00:10	0:23:00	0:25:50
LA19	15:40:00	0:00:05	15:40:00	0:00:22	0:00:00	- 0:00:22
LA21	21:04:00	0:00:06	20:09:00	0:01:55	0:55:00	0:53:05
LA27	24:02:00	0:00:06	24:02:00	0:06:11	0:00:00	- 0:06:11
LA30	25:34:00	0:00:07	25:23:00	0:05:47	0:11:00	0:05:13
LA40	23:54:00	0:00:21	23:00:00	0:05:14	0:54:00	0:47:46

Z naměřených výsledků je patrné že v 4 z 10 případů (tab. 6-25 žlutě) je časová investice do optimalizace pomocí LS zbytečná vzhledem k tomu, že LS není schopný uniknout z lokálního extrému daného počátečním řešením. Za přijatelné výsledky pak lze považovat časový přínos u modelů FT10 a LA2, kde byla úspora času cca půl hodinová. Takto získaný čas lze využít pro kompenzaci nenadálých událostí, jako jsou poruchy atd. Velmi dobrého výsledku bylo pak dosaženo u modelů LA21 a LA40, u kterých úspora (časový přínos) dosáhla bez mála hodinu.

Tab. 6-26: Porovnání časového přínosu optimalizace LS - FJSSP

Model	Nejlepší řešení CA [hod:min:sek]	Čas CA [hod:min:sek]	Nejlepší řešení LS [hod:min:sek]	Čas LS [hod:min:sek]	Zlepšení [hod:min:sek]	Přínos [hod:min:sek]
MK01	0:47:00	0:00:02	0:45:00	0:00:07	0:02:00	0:01:53
MK02	0:38:00	0:00:02	0:36:00	0:00:19	0:02:00	0:01:41
MK03	3:24:00	0:00:06	3:24:00	0:02:49	0:00:00	-0:02:49
MK04	1:17:00	0:00:03	1:17:00	0:00:40	0:00:00	-0:00:40
MK05	3:09:00	0:00:03	3:09:00	0:00:47	0:00:00	-0:00:47
MK06	1:33:00	0:00:06	1:33:00	0:01:51	0:00:00	-0:01:51
MK07	3:20:00	0:00:04	3:20:00	0:01:51	0:00:00	-0:01:51
MK08	8:43:00	0:00:08	8:43:00	0:00:08	0:00:00	-0:00:08
MK09	5:49:00	0:00:09	5:49:00	0:07:26	0:00:00	-0:07:26
MK10	4:26:00	0:00:11	4:26:00	0:00:11	0:00:00	-0:00:11

Jak je patrné z předešlých analýz, navržený algoritmus LS není příliš vhodný pro testované matematické modely FJSSP. Pouze ve dvou případech (tab. 6-26 přínos - bílé) je dosaženo časové úspory a navíc velmi malé. V ostatních případech není nalezen lepší výsledek, než je počáteční řešení dané CA.

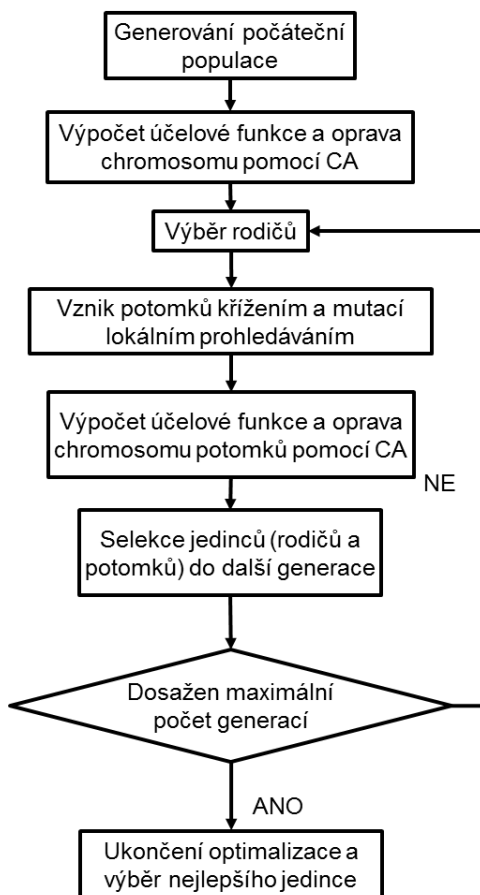
Celkově jsou úspory času pomocí LS velmi relativní. Skutečný přínos lze zhodnotit pouze v závislosti na frekvenci odvádění hotové výroby. V případě, že se výroba odvádí ihned po dokončení operace, lze takto vytvořenou úsporu využít jak na neurčitosti ve výrobě, tak pro zařazení dalších zakázek do výroby.

Porovnání přínosů bylo provedeno konverzí obecných časových jednotek na minuty. Pokud by byly převedeny na hodiny, přínos pomocí LS by byl daleko větší.

Posledním kritériem pro volbu optimalizace pomocí LS je dostupný čas pro optimalizaci v případě, že se nejedná o přepřítávání s ohledem na změny ve výrobním systému (chybějící materiál, defekty strojů a výrobku atd.). Tehdy lze přistoupit k optimalizaci s vědomím, že existuje možnost nalezení lepšího řešení než stávajícího, vygenerovaného pomocí prioritního pravidla.

6.3 Evoluční algoritmus

V kapitole č. 4 byl zvolen jako pokročilý optimalizační nástroj evoluční algoritmus. Vyvinutý algoritmus pro účely testování modelů vychází ze základní filozofie genetických algoritmů (obr. 6-3) avšak obsahuje prvky, které vycházejí v oblasti selekce z evolučních strategií a umělé inteligence.



Obr. 6-3: Diagram evolučního algoritmu

Následující kapitoly se zabývají vývojem jednotlivých kroků navrhovaného evolučního algoritmu.

V těchto kapitolách je použito následující názvosloví:

- Populace – množina řešení iteračního kroku.
- Rodič – řešení použité pro generování potomků pomocí křížení a mutace.
- Potomek – nově vygenerované řešení.
- Generace – iterační krok algoritmu.
- Chromosom – reprezentace daného problému, v tomto případě řetězec určující pořadí zakázek resp. operací na strojích.
- Sub-chromosom – část chromosomu ohraničená její působností, například vazbou na jeden stroj.
- Gen – část chromosomu určující pořadí právě jedné zakázky (operace) na daném stroji.

6.3.1 Volba reprezentace

Pro použití evolučního algoritmu je nutné reprezentovat námi řešený problém (data o něm) ve formě chromosomu. V našem případě je to způsob zápisu posloupnosti operací na strojích do struktury, umožňující provádění výpočtů nutných k tvorbě a hledání řešení.

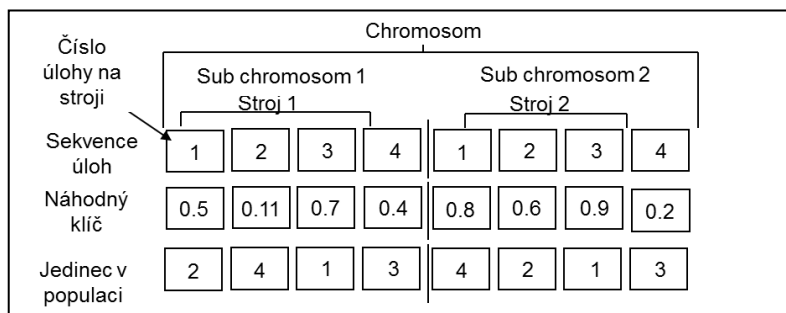
Pro reprezentování rozvrhovacího problému se nejčastěji používají tyto reprezentace [153]:

- reprezentace založená na úlohách,
- reprezentace založená na operacích,
- reprezentace preferenčním seznamem,
- reprezentace disjunktivním grafem,
- reprezentace založená na vzájemném vztahu mezi pracemi,
- reprezentace založená na prioritních pravidlech,
- reprezentace založená na časech dokončení,
- reprezentace založená na strojích,
- reprezentace pomocí náhodných klíčů.

Tyto reprezentace mohou být rozděleny podle kódovacího postupu na:

- reprezentace s přímým přístupem (seznam jednotlivých operací),
- reprezentace založená na operacích, reprezentace založená na vzájemném vztahu mezi operacemi, reprezentace založená na časech dokončení a reprezentace náhodných klíčů,
- reprezentace s nepřímým přístupem (seznam jednotlivých prioritních pravidel, podle kterých uspořádáme operace),
- reprezentace založená na preferenčním seznamu, reprezentace založená na prioritních pravidlech, reprezentace založená na disjunktivním grafu a reprezentace založená na strojích.

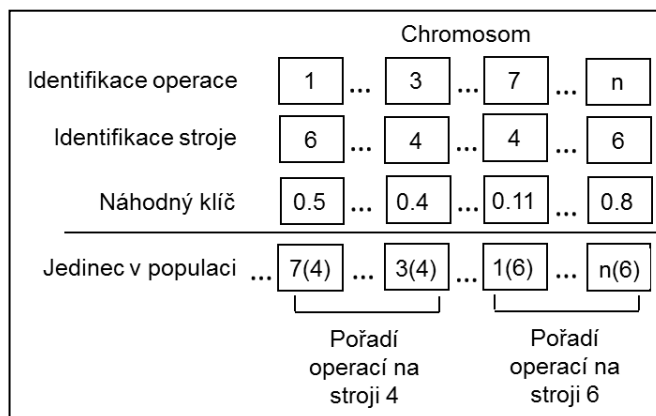
Populace je v tomto algoritmu založena na reprezentování chromosomu náhodným klíčem popsané Beanem [154]. Tzn. náhodně generovaná čísla jsou použita k tvorbě chromosomu jedinců resp. seřazení genů (operací) uvnitř subchromosomů k vytvoření nového řešení daného problému (obr. 6-4).



Obr. 6-4: Reprezentace pomocí náhodného klíče

Přestože tato reprezentace nebývá v řešení teoretických problémů používána na rozdíl od reprezentace založené na úlohách či operacích, je vyhledávána zvláště pro svoji robustnost. Je schopná sama redukovat chyby při křížení a mutaci a lze pomocí ní řešit v rámci jednoho křížení a mutace více druhů omezení. Proto je používána zvláště pro reálné rozvrhovací problémy i přes to, že neposkytuje v rámci křížení takové možnosti jako ostatní reprezentace [155].

Tuto reprezentaci lze použít i v souvislosti s omezením flexibility a tedy rozvrhovacího problému FJSSP. V takovém případě je však nutné přiřadit reprezentaci volby stroje a modifikovat použití klíče pro tvorbu jedince (řešení) v populaci (viz obr. 6-5).



Obr. 6-5: Reprezentace pomocí náhodného klíče pro JSSP a FJSSP

V kroku generace počátečního řešení je pak dle této formy reprezentace vygenerována první populace náhodně.

6.3.2 Oprava chromosomu

Zde je stejně jako pro výpočet funkce fitness využit konstruktivní algoritmus založený na principu algoritmu Gifflera a Thompsona. Hlavní rozdíl oproti tomuto algoritmu je v bodě 3 jeho popisu, který zajišťuje vlastní opravu (posunutí operace na vyhovující místo v pořadí).

Pomocí náhodného generování nové populace a při nevhodné manipulaci s geny pomocí křížení a mutace vznikají řešení, která nesplňují požadavky aktivních rozvrhů či rozvrhů bez zpoždění. Z tohoto důvodu je do konstruktivního algoritmu implementován mechanismus opravy chromosomu.

Tato oprava nahrazuje obvyklejší penalizaci či výmaz jedince, který je nahrazen novým náhodně vygenerovaným řešením.

Jednotlivé kroky algoritmu jsou popsány pro aktivní rozvrhy, avšak lze je použít stejným způsobem i pro rozvrhy bez zpoždění.

t - krok

P_t – částečný rozvrh ($t-1$) rozvržených operací

R_t – sekvence operací na stroji generovaných náhodným klíčem

S_t – množina operací rozvrhovatelých v kroku t , tzn. všechny operace, které musí předcházet operacím v S_t jsou v P_t

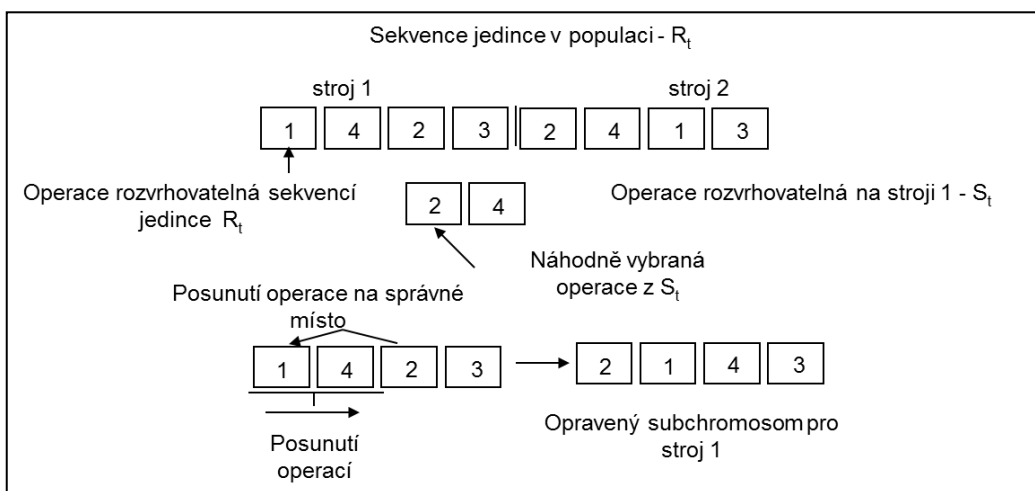
e_t – nejkratší čas, kdy operace o_k v S_t může začít

f_t – nejkratší čas kdy operace o_k v S_t může být ukončena, kde $f_t = o_k + p_k$, kde p_k je procesní čas o_k .

Následují samotné kroky algoritmu:

1. Necht' $t = 1$ s prázdným rozvrhem P_t . S_t je množina všech operací bez předchůdců (ty, které jsou první v úloze) a R_t je množina všech operací, které jsou první v sekvenci generované náhodným klíčem.
2. Nalezni $f_t^* = \min_{o_k \in S_t} \{f_t\}$ a stroj M^* , ke které přísluší f_t^* . Pokud je více možností výběru M^* , zvolí se libovolný z nich.
3. Pokud je operace o_k v obou množinách R_t a S_t , které
 - I. vyžadují M^*
 - II. $e_t < f_t^*$
 vyber o_k z R_t .

Pokud operace není na správné pozici v R_t , vyber o_k z S_t náhodně a oprav sekvenci posunutím operace na odpovídající pozici v R_t a posuň ostatní operace v R_t (obr. 6-6).



Obr. 6-6 : Oprava chromosomu

V případě, že se operace nevyskytuje na daném stroji, je tato nalezena v sekvenci ještě nerozvrhnutých operací na ostatních strojích. Následně je vyjmuta a vložena do sekvence příslušného stroje.

4. Posun na další krok:
 - I. Přidání o_j do P_t , tzn. tvorba P_{t+1} .
 - II. Smazání o_j z S_t a tvorba S_{t+1} přidáním operace, která bezprostředně následuje o_j v S_t v dané úloze (pokud o_j není poslední v dané úloze).
 - III. Posun na další operaci v R_t na daném stroji do R_{t+1} .
 - IV. Zvýšení t o 1.

5. Pokud zbývají nějaké nerozvržené operace ($t < nm$), zpět do bodu 2. V opačném případě se algoritmus ukončí.

Tento způsob opravuje chromosom tak, aby byla co nejvíce zachována sekvence daná křížením či mutací.

Po opravení sekvence je opravena i reprezentace jedince náhodným klíčem tak, aby popisovala sekvenci, která byla využita pro výpočet funkce fitness.

6.3.3 Selektce

Operaci selektce lze rozdělit na dvě oblasti a to na selekci rodičů pro vznik nových řešení a selekci rodičů a potomků do další generace.

V oblasti selektce rodičů pro vznik potomků je uplatněno tzv. ruletové pravidlo, kdy na základě hodnoty účelové funkce je přidělena pravděpodobnost $P(x_i)$ (19), že bude rodič vybrán pro křížení (pravděpodobnost určena pro minimalizaci funkce, kde N je velikost populace).

$$P(x_i) = \frac{\sum_{j=1}^N f_j - f(x_i)}{\sum_{j=1}^N f_j} \quad (19)$$

U tohoto způsobu je známo, že dochází vlivem výběru ke stagnaci populace resp. přílišné konvergenci k nejlepšímu řešení a tím uvážnutí v lokálním extrému. Z tohoto důvodu je střídavě vybírán na základě pravděpodobnosti nejlepší a nejhorší jedinec, aby v rámci jednotlivých generací nedocházelo k zániku potenciálně vhodného řešení reprezentovaného chromosomem. Pravděpodobnost (20) výběru druhého (resp. sudého) rodiče je pak

$$P(x_i) = \frac{f(x_i)}{\sum_{j=1}^N f_j} \quad (20)$$

Tímto způsobem je vybráno N rodičů, kteří generují N potomků. Neznamena to však, že jsou vybráni všichni jedinci z celé populace, pouze nejúspěšnější a nejhorší jedinci mají určitou pravděpodobnost, že budou vybráni vícekrát.

Druhá oblast selektce je výběr jedinců do nové populace. Do vyvíjeného algoritmu jsou implementovány tři způsoby:

- Elitářský přístup (E).
- Elitářský přístup s kontrolou klonů (ECC).
- Turnajový přístup (PvCH).

Elitářský přístup

V tomto způsobu se stará populace N_1 a nově vniklí potomci N_2 sloučí do výběrové populace N^* , ve které jsou seřazeni jedinci na základě ohodnocení účelové funkce bez rozdílu, zda se jedná o jedince z N_1 či N_2 .

Elitářský přístup s kontrolou klonů

Elitářský přístup často zapříčiňuje rychlou konvergenci populace k nejlepšímu jedinci a při malé pravděpodobnosti křížení, která je obvykle nastavená na 10%-20%, dochází v kombinaci s principem opravy chromosomu k vzniku identických jedinců tzv. klonů.

Pro zamezení výskytu tzv. klonů byl vytvořen jednoduchý algoritmus, který vyhledává identické jedince v populaci a pokud jejich počet přesáhne číslo definované např. v závislosti na velikosti populace či fixně, je jedinec nahrazen. V tomto případě se považuje klon za jedince se stejnou hodnotou účelové funkce. Vzniká tak sice nebezpečí odstranění řešení s jinou reprezentací chromosomu, avšak vzhledem k velikosti rozvrhovacích problémů by kontrola každého genu mezi jedinci byla časově náročná.

Samotné nahrazení probíhá v rámci souboje nových a starých jedinců, kdy takto nalezení jedinci jsou nahrazováni postupně od nejhorších k nejlepším jedincům ze soubojové množiny N^* .

Touto procedurou se dává taktéž možnost i nejhorším jedincům se reprodukovat s vědomím, že pro vytvoření nejlepších jedinců je často nutné provést malou změnu na těch nejhorších.

Turnajový přístup

Tato evoluční strategie využívá znalosti o příslušnosti potomků k rodičům. U každého potomka je sledována příslušnost vzhledem k rodiči, která vychází z operace křížení. Následně v operaci souboje jsou porovnávány hodnoty účelové funkce jednotlivých rodičů a jeho potomků. Rodič pak zůstává v nové generaci, pokud není nahrazen lepším potomkem, než je on sám.

Tímto způsobem se ve velké míře zamezí vzniku klonů. Následuje procedura totožná jako v elitářském přístupu, avšak seřazení již neprobíhá v rámci stávající populace a nových jedinců, ale až v rámci nově vzniklé populace. Touto procedurou je připravena nová generace na nový výběr jedinců pro křížení.

6.3.4 Křížení

Křížení společně s mutací patří mezi operátory, kteří jsou odpovědní za vznik nových jedinců (potomků) z původních řešení (rodičů). Operátory křížení lze rozdělit na základní dvě skupiny a to specifické (např. křížení založené na operacích), které lze použít pouze v kombinaci s určitým druhem reprezentace problému, a obecné, které lze použít na většinu známých reprezentací (např. jednobodové).

Pro vyvíjený algoritmus vzhledem ke zvolené reprezentaci je zvolen obecný typ křížení a to uniformní křížení. Toto křížení je zvoleno na základě vlastností reprezentace náhodným klíčem, aby podpořilo robustnost daného algoritmu tzn., aby jak samotná reprezentace, tak i křížení bylo použitelné nejenom pro klasický problém určování sekvence operací na daných strojích, ale i pro ostatní problémy v rozvrhování jako je určení stroje v případě, kdy je možné operaci provést na různých pracovištích.

Uniformní křížení na základě definované pravděpodobnosti (resp. koeficientu křížení, který udává pravděpodobnost zachování genu v jedinci) a pravděpodobnostního klíče určuje, zda dojde k výměně jednotlivých genů (obr. 6-7).

		Chromosom									
Rodič 1 náhodný klíč		0,55	0,34	0,86	0,27	...	0,44	0,65	0,91	0,12	
Stroj		1	5	4	8	...	2	8	6	4	
Rodič 2 náhodný klíč		0,23	0,7	0,35	0,64	...	0,8	0,37	0,1	0,58	
Stroj		3	8	5	9	...	1	4	5	4	
Pravděpodobnostní klíč		0,6	0,3	0,81	0,45	...	0,53	0,95	0,7	0,61	
Potomek 1 náhodný klíč		0,55	0,34	0,35	0,27	...	0,44	0,37	0,91	0,12	
Stroj		1	5	5	8	...	2	4	6	4	
Potomek 2 náhodný klíč		0,23	0,7	0,86	0,64	...	0,8	0,65	0,1	0,58	
Stroj		3	8	4	9	...	1	8	5	4	

Obr. 6-7: Ukázka uniformního křížení

Přes to, že je reprezentace náhodným klíčem považována za robustní reprezentaci, dochází v některých případech k narušení chromosomu genem se stejnou hodnotou klíče.

K tomuto poškození dochází v případě, že se kříží příbuzenská řešení (například rodič a potomek). V rámci tohoto incestu je možná výměna genu, který obsahují obě řešení a způsobí tak nemožnost řádného seřazení operací podle vzniklého chromosomu. Proto je do operace křížení implementován mechanismus, který nedovolí výměnu genu, jehož hodnotu obsahují obě řešení určená k reprodukci.

6.3.5 Mutace

Mutace je v tomto algoritmu zastoupena v několika oblastech. První oblast zajišťující princip mutace je prezentovaná oprava chromosomu, kdy u poškozených chromosomů dochází k opravě genů, které nevyhovují danému druhu rozvrhu. Druhým mechanismem zaručujícím různorodost chromosomu v populaci je princip selekce, tzv. kontrola klonů. Posledním mechanismem je klasická mutace, která je aplikována v rámci křížení na definovaném procentu vybraných rodičů.

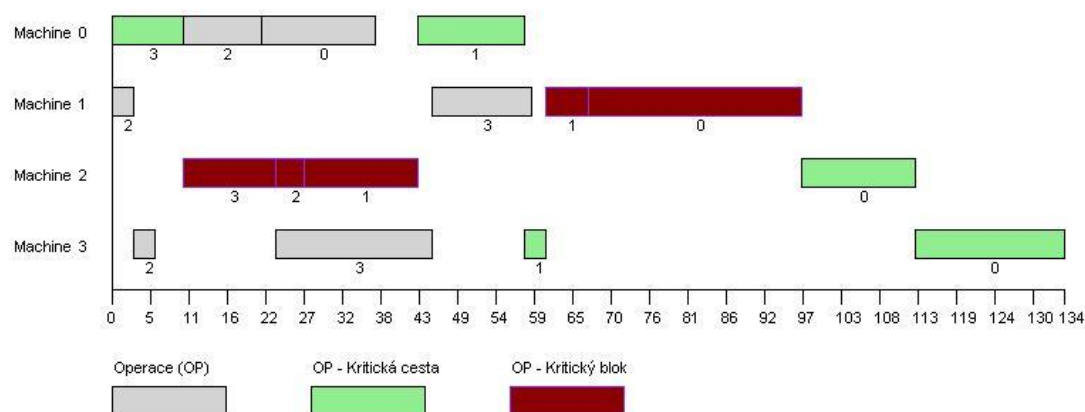
Jedná se o hybridizaci evolučního algoritmu lokálním prohledáváním, kde jsou vybrané geny prohozeny (je zaměněna sekvence operací) pomocí znalosti o kritické cestě.

Kritická cesta (CP – critical path) je termínem určujícím nejdelší cestu od počátku procesu k jeho konci. Tzn. kritická cesta zakázky je tedy taková posloupnost operací, pro kterou platí, že pokud dojde ke skluzu na některé z nich, dojde zároveň i ke skluzu celé zakázky. Kritický blok je pak „blok“ operací na kritické cestě, zpracovávané za sebou na jednom stroji.

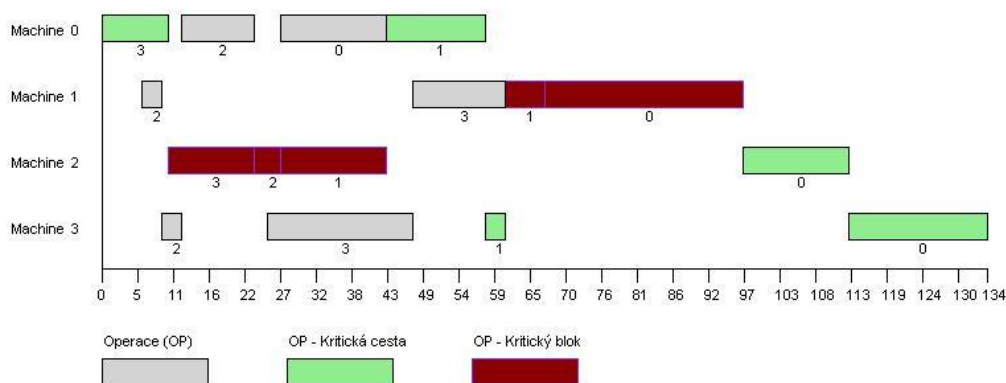
Pro výpočet kritické cesty resp. kritických bloků bývá využívána tzv. metoda kritické cesty (CPM - critical path method). Pro řešení metodou kritické cesty se

nejčastěji využívá tzv. síťový graf. Pro analýzu kritické cesty (CPA - Critical Path Analysis) byl modifikován algoritmus Gifflera a Thompsona.

Nejprve je rozvržen celý rozvrh dopředně (obr. 6-8) a jsou zaznamenány nutné informace o operacích (počáteční a koncové časy operací) a samotném plánu (sekvence operací na strojích, makespan). Následovně je tento rozvržen pomocí zaznamenaných informací zpětně (obr. 6-9). Analýza kritické cesty pak porovnává jednotlivé počáteční resp. koncové časy operací obou plánů a tam, kde jsou shodné, je operace na kritické cestě. Zároveň je vyhodnocováno, zda mají operace na kritické cestě a shodném stroji totožný čas ukončení jako zahajovací čas následující operace na téže stroji pro určení tzv. kritických bloků.



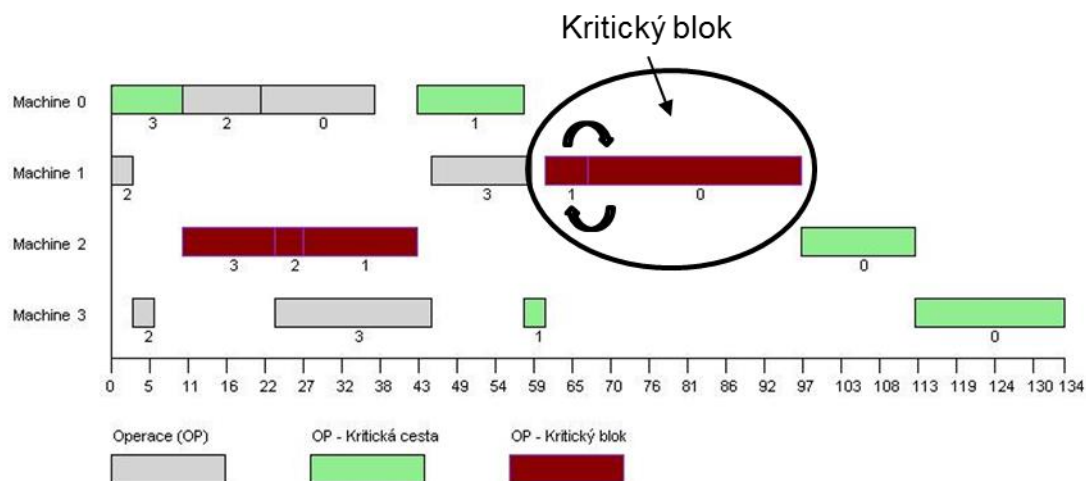
Obr. 6-8: Dopředný rozvrh



Obr. 6-9: Zpětný rozvrh po fixaci na dosažený makespan

Znalost, které operace se nachází v kritických blocích, je pak využita pro lokální prohledávání.

Jak je uvedeno v kapitole 4.4.2, je výkonnost a rychlost procedury lokálního prohledávání závislá na prohledávaném okolí (sousedství). V rámci tohoto algoritmu je pak aplikován princip Nowicki E., Smutnicki C. [116], kde po CPA je náhodně vybrán jeden z kritických bloků, následně je vybrána první nebo poslední dvojice operací v kritickém bloku a na těchto operacích je provedena záměna sekvence (obr. 6-10).



Obr. 6-10: Změna sekvence operací na kritickém bloku

6.3.6 Volba nastavení operátorů velikost populace, selekce a křížení

Důležitým krokem při aplikaci evolučního algoritmu je nastavení jeho parametrů. Tyto parametry ovlivňují jak délku optimalizace, tak dosažené hodnoty účelové funkce (v tomto případě minimalizace makespan). V rámci testování těchto parametrů je vybrán jeden zástupce z každého druhu modelů. V případě modelů JSSP je to FT10, který je považován již dlouhou dobu za nejobtížnější JSSP problém [156], [157]. U modelů FJSSP byl pak vybrán MK10 jako problém, který obsahuje nejvíce obtížné řešitelných lokálních extrémů [158].

Mezi parametry, které ovlivňují dosažený výsledek a optimalizační čas navrženého algoritmu, patří:

- Velikost populace – zásadním způsobem ovlivňuje délku optimalizace a výslednou hodnotu účelové funkce [159]. Velikost populace definuje počet jedinců, tedy kolik řešení v jedné iteraci algoritmu bude vypočteno. Počet jedinců v jedné populaci také ovlivňuje konvergenci populace k jednomu řešení. V případě, že je populace příliš malá, dochází ke konvergenci k jednomu řešení (uvíznutí algoritmu v lokálním extrému). Pokud je naopak populace příliš velká, je malá pravděpodobnost výběru vhodných jedinců pro následnou reprodukci, tzn. populace se vyvíjí velmi pomalu [160]. Nejčastěji se objevují dva přístupy. V prvním přístupu se volí fixní velikost populace bez ohledu na velikost reprezentace (např. délku chromosomu) [161]. V případě druhém pak velikost populace odpovídá násobku délky chromosomu (např. počtu rozvrhovaných operací) [162]. Pro testování tohoto algoritmu byl zvolen druhý způsob (19) a tedy

$$N = CH * 2 + 100 \quad (19)$$

kde N je velikost populace a CH počet operací.

- Počet generací – počet generací (počet iterací algoritmu) ovlivňuje jak výpočetní čas, tak výslednou hodnotu účelové funkce. V tomto případě platí, že čím větší je počet

iterací, tím má populace delší dobu na vývoj a tudíž je hodnota účelové funkce blíže optimu. Vývoj populace však není lineární. V prvních generacích se populace obvykle vyvíjí rychle, později spíše stagnuje s ohledem na vyšší výskyt lokálních extrémů. Pro navržený EA je zvolen počet iterací 200 i s ohledem na předpokládanou délku optimalizace [163].

- Koeficient (pravděpodobnost) křížení – ovlivňuje procento sdílení genů v rámci chromosomu a tedy výslednou hodnotu účelové funkce. Pokud je pravděpodobnost příliš malá, dochází pouze k malým změnám a proces připomíná spíše princip lokálního prohledávání a je tedy možné předpokládat uvíznutí populace v lokálním extrému. Při příliš velké pravděpodobnosti křížení lze předpokládat, že algoritmus se bude chovat spíše jako náhodné prohledávání. Navržený algoritmus je proto testován v rozsahu parametru křížení 0,6-0,9 resp. 40%-10% pravděpodobností, že dojde k výměně genu.
- Selekce jedinců do nové populace – ovlivňuje konvergenci populace a tedy hodnotu účelové funkce. V rámci navrženého algoritmu jsou testovány tři základní principy:
 - Elitářský (E)
 - Elitářský s kontrolou klonů (ECC)
 - Turnajový (PvCH)
- Koeficient (rozsah) hybridizace lokálním prohledáváním – resp. mutace ovlivňuje výslednou hodnotu účelové funkce a délku optimalizace. Pomocí hybridizace dojde k změně pořadí operací na stroji v rámci jednoho jedince a tím k přiblížení řešení k optimální hodnotě účelové funkce. Na druhou stranu, čím větší bude rozsah hybridizace, tím bude delší optimalizační čas – při hybridizaci probíhá výpočet dopředného a zpětného rozvrhu. Pro testování hybridizace jsou stanoveny rozsahy parametru hybridizace 0,05-1 (na 5%-100% populace bude provedena hybridizace).

V následujících krocích jsou provedeny analýzy vlivu výše popsanych parametrů v prostředí statistického softwaru MiniTab a MS Excell (tab. 6-27).

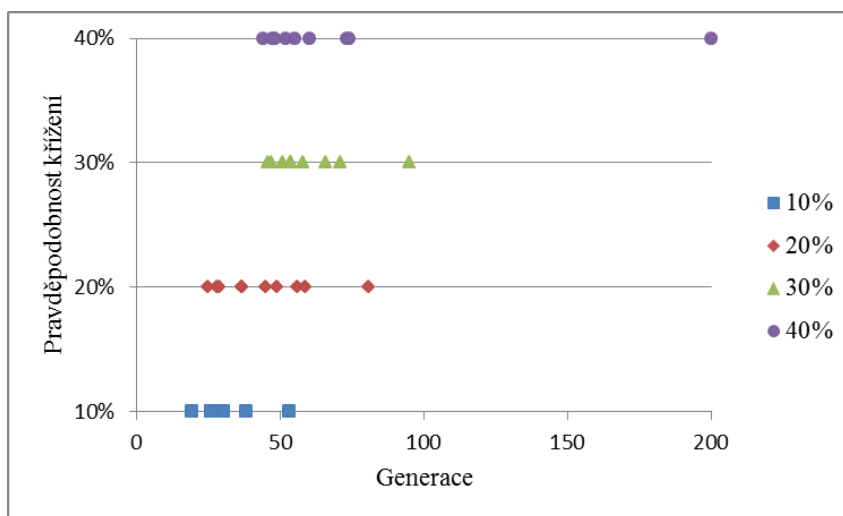
Tab. 6-27: Nastavení parametrů EA v jednotlivých analýzách

Test	Velikost populace	Počet generací	Pravděpodobnost křížení	Druh selekce	Počet opakování při jednom nastavení	Celkový počet experimentů
Analýza vlivu parametru křížení na konvergenci populace k jednomu řešení.	Model FT10 300 Model MK10 700	200	10%,20%,30%,40%	E	10	80
Analýza vlivu parametru křížení a selekce na hodnotu účelové funkce a potřebný výpočetní čas.	Model FT10 300 Model MK10 700	200	10%,20%,30%,40%	E, ECC, PvCH	10	240

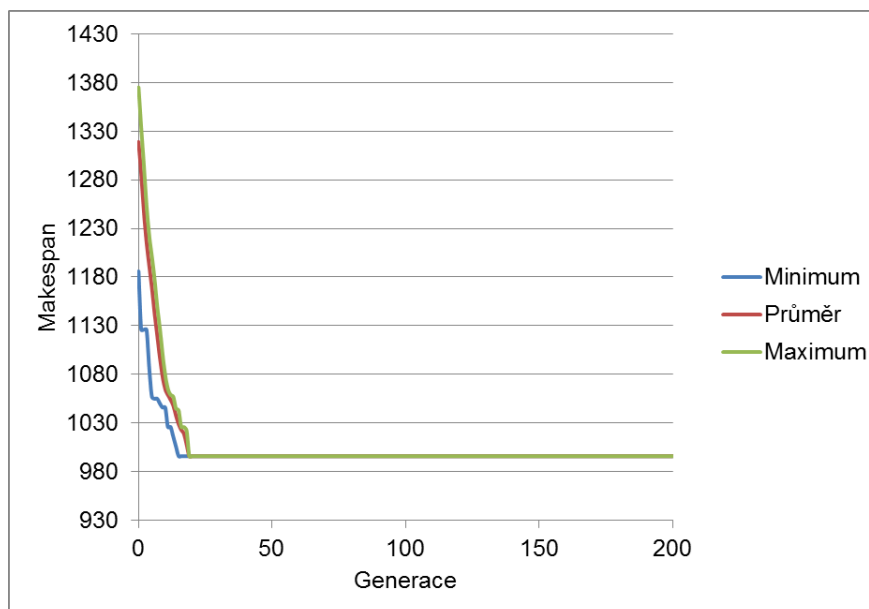
V prvním kroku je provedeno testování, na kolik ovlivňuje pravděpodobnost křížení konvergenci populace k lokálnímu extrému. Pro tento test je zvolen elitářský přístup selekce jedinců do další generace vzhledem k tomu, že neobsahuje žádný mechanismus, který by reguloval konvergenci populace.

Z výsledků testů (graf 6-1) je patrné, že čím větší je pravděpodobnost křížení, tím celá populace konverguje v pozdějších generacích. U 10% křížení populace

konvergovala nejrychleji, v jednom případě dokonce v 19. generaci (graf 6-2 zobrazuje vývoj minimální, průměrné a maximální hodnoty účelové funkce).

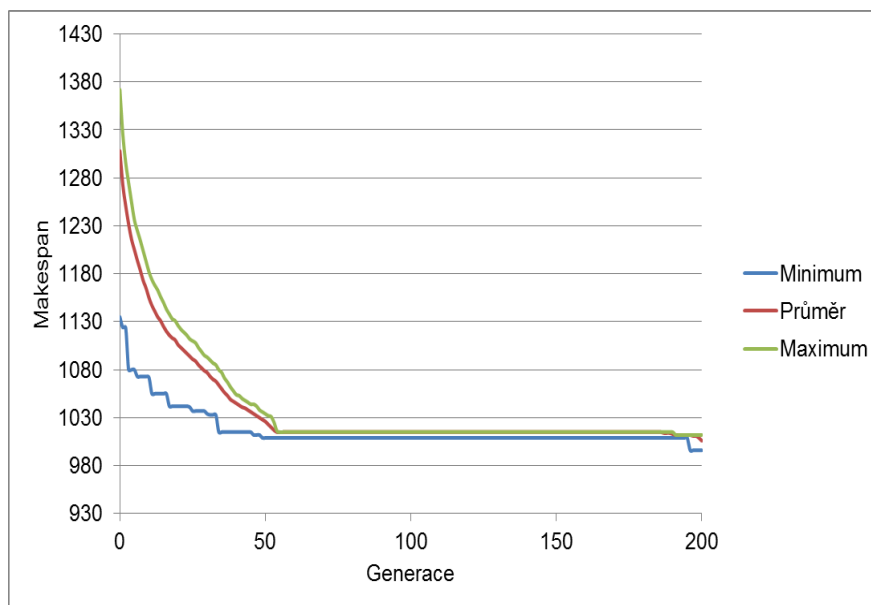


Graf 6-1: Model FT10 konvergence celé populace k jednomu řešení



Graf 6-2: Model FT10 – ukázka konvergence celé populace k jednomu řešení při 10% pravděpodobnosti křížení

Při použití 40% křížení konvergovala populace nejpomaleji. V jednom z případů populace za 200 generací nekonvergovala vůbec (graf 6-3).



Graf 6-3: Model FT10 ukázka populace u které nedošlo ke konvergenci všech jedinců při 40% pravděpodobnosti křížení

Důležitým hlediskem pro volbu pravděpodobnosti křížení je výsledná hodnota účelové funkce a čas potřebný pro optimalizaci. Tab. 6-28 zobrazuje minimální, maximální a průměrné hodnoty z 10 měření modelu FT10 (optimální hodnota FT10 - 930)

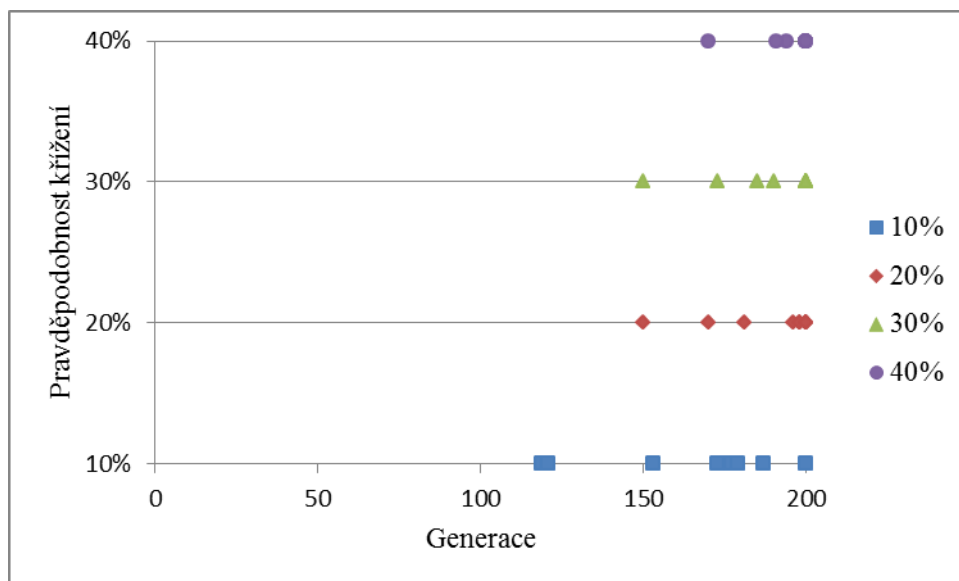
Tab. 6-28: Výsledky testování FT10 jednoduchý elitismus

FT10 - Jednoduchý elitismus (930)				
Pravděpodobnost křížení	10%	20%	30%	40%
Minimum $f(x)$	996	991	1013	988
Maximum $f(x)$	1049	1015	1030	1027
Průměr $f(x)$	1024,1	1006	1016,5	1012,4
Průměrný čas optimalizace [hh:mm:ss]	0:09:23	0:09:25	0:10:29	0:10:33

Z výsledků je patrné, že hodnota účelové funkce je značně ovlivněna typem selekce, kdy 40% pravděpodobnost dosáhla nejlepšího řešení. Toto řešení je však dosaženo v pozdějších generacích oproti nastavení 10% křížení, které předčasně konvergovalo.

Čas optimalizace s rostoucí pravděpodobností mírně roste. Rozdíly mezi délkami optimalizace jednotlivých úrovní nastavení parametrů jsou však natolik malé, aby to ovlivnilo volbu křížení v praktických aplikacích.

Konvergence celé populace u druhého testovaného modelu MK10 má podobný charakter, přes to, že jí je dosaženo mnohem později. V mnohých případech populace nekonvergovala k jednomu řešení (viz graf 6-4).



Graf 6-4: Model MK10 konvergence celé populace k jednomu řešení

Výsledky testování modelu MK10 potvrzují, že 10% křížení dosahuje lepších výsledků, pokud celá populace nekonverguje k jednomu řešení. Toto nastavení nejenže dosáhlo nejlepšího výsledku, ale také průměrná a maximální hodnota účelové funkce je ze všech nastavení pravděpodobnosti nejlepší (tab. 6-29). Časy optimalizace mezi jednotlivými nastaveními se jako v předešlém případě o mnoho neliší, je zde ale patrný rozdíl mezi jednotlivými modely. Rozdíl v optimalizačních časech je způsoben jednak počtem operací v rozvrhu, ale také počtem jedinců v populaci od nich odvozených.

Tab. 6-29: Výsledky testování MK10 jednoduchý elitismus

MK10 - Jednoduchý elitismus (165)				
Pravděpodobnost křížení	10%	20%	30%	40%
Minimum $f(x)$	287	297	301	290
Maximum $f(x)$	298	304	307	308
Průměr $f(x)$	294,6	301,8	304,2	303,4
Průměrný čas optimalizace [hh:mm:ss]	1:51:00	1:54:56	1:55:34	1:42:56

V dalším kroku je testováno na modelech FT10 (tab. 6-30) a MK10 (tab. 6-31), jakým způsobem ovlivní hodnotu účelové funkce použití mechanismů pro úpravu konvergence pomocí kontroly klonů a turnajové selekce při různém nastavení pravděpodobnosti křížení.

Tab. 6-30: Výsledky testování FT10 ECC

FT10 - Elitismus s kontrolou klonů (930)				
Pravděpodobnost křížení	10%	20%	30%	40%
Minimum $f(x)$	970	990	1000	988
Maximum $f(x)$	1013	1022	1028	1020
Průměr $f(x)$	997,2	1010,2	1010,6	1007,4
Průměrný čas optimalizace [hh:mm:ss]	0:09:02	0:09:15	0:09:16	0:09:16

Tab. 6-31: Výsledky testování MK10 ECC

MK10 - Elitismus s kontrolou klonů (165)				
Pravděpodobnost křížení	10%	20%	30%	40%
Minimum $f(x)$	238	270	264	277
Maximum $f(x)$	268	287	280	293
Průměr $f(x)$	255,3	278,9	271,2	286,2
Průměrný čas optimalizace [hh:mm:ss]	1:50:41	1:47:09	1:47:55	1:42:47

Při použití mechanismu kontroly klonů (tab. 6-30 a 6-31) dosahuje 10% pravděpodobnost nejlepších výsledků s ohledem na minimální, maximální i průměrné hodnoty a to v případě obou modelů. Navržená metoda úpravy konvergence (ECC) vykazuje lepší výsledky než jednoduchý elitismus (E).

Čas trvání optimalizace použitím ECC je neznatelně nižší, než tomu je u selekce E. To je patrně způsobeno nižším počtem oprav chromosomu. Pravděpodobnost křížení sousedících jedinců vlivem většího rozložení populace po množině všech řešení generuje více reálných rozvrhů. Dále je to také způsobeno větší stratifikací hodnot účelové funkce a tudíž rychlejším řazením populace pomocí $f(x)$.

Dalším způsobem úpravy konvergence je navržená turnajová selekce rodičů s dětmi (PvCH tab. 6-32 a 6-33).

Tab. 6-32: Výsledky testování FT10 PvCH

FT10 - Turnaj PvCH (930)				
Pravděpodobnost křížení	10%	20%	30%	40%
Minimum $f(x)$	1000	1015	1015	1013
Maximum $f(x)$	1024	1032	1023	1015
Průměr $f(x)$	1015,6	1022,4	1017,5	1014,4
Průměrný čas optimalizace [hh:mm:ss]	0:08:56	0:09:00	0:09:00	0:09:01

Tab. 6-33: Výsledky testování MK10 PvCH

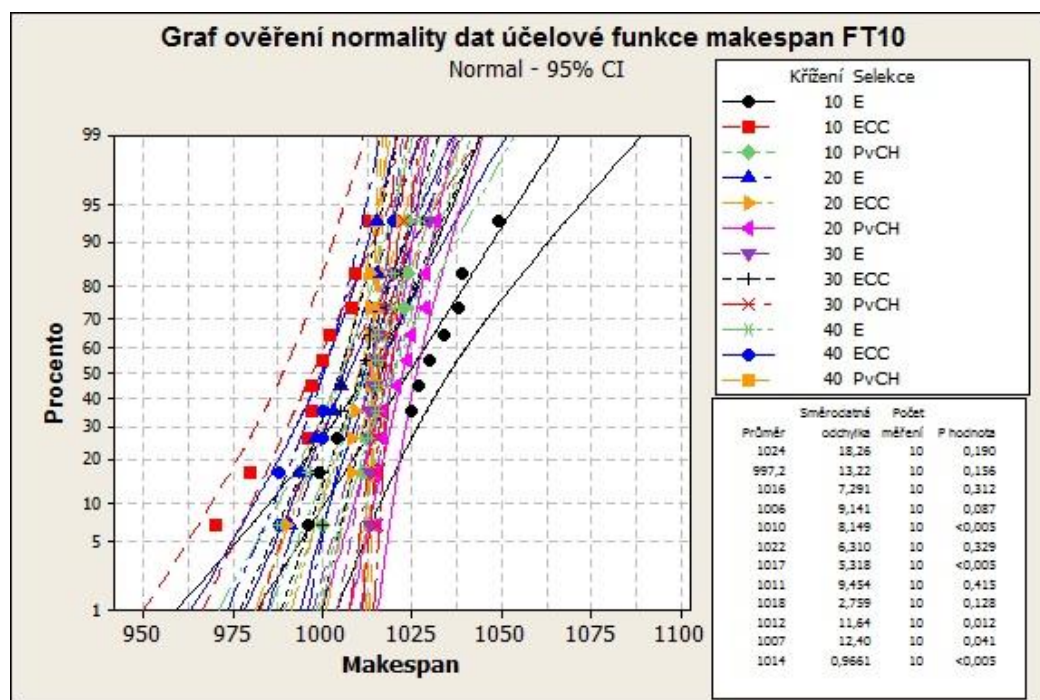
MK10 - Turnaj PvCH (165)				
Pravděpodobnost křížení	10%	20%	30%	40%
Minimum $f(x)$	290	289	300	300
Maximum $f(x)$	306	309	315	317
Průměr $f(x)$	298,2	300,7	306,6	310
Průměrný čas optimalizace [hh:mm:ss]	1:42:56	1:42:51	1:39:55	1:33:15

Přestože tento způsob je nepatrně méně časově náročný na optimalizaci, vykazuje horší hodnoty účelové funkce než metoda ECC a dokonce E. To je patrně způsobeno samotným principem metody, kdy na rozdíl od ECC je pravděpodobnost křížení se svým potomkem takřka nulová. V PvCH jsou rodiče porovnáváni se svými nejvíce podobnými potomky z hlediska hodnot chromosomu na rozdíl od ECC, kde jsou podobná řešení vylučována z populace pouze na základě hodnoty účelové funkce.

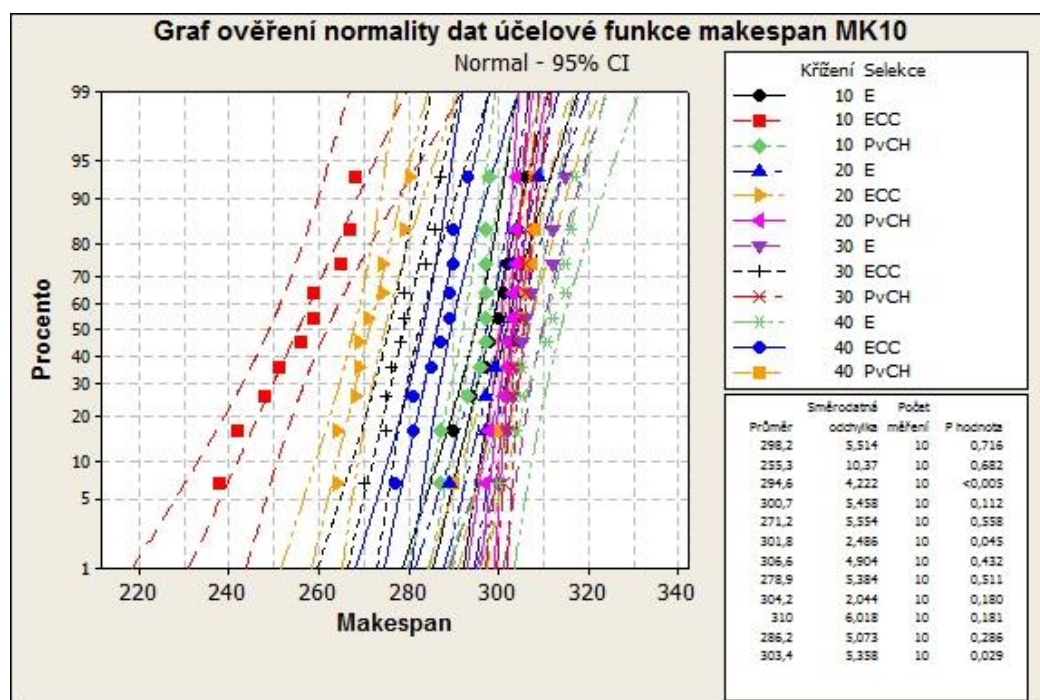
Z počáteční analýzy je patrné, že metoda ECC spolu s 10% křížením poskytuje nejlepší výsledky. Tyto výsledky však mohou být ovlivněny interakcí mezi nastavením parametrů selekce a křížení, což je ověřeno pomocí následujících testů. Prvním krokem při ověření interakce mezi parametry křížení a selekce je analýza normality a

směrodatných odchylek pro jednotlivé modely a jejich nastavení (graf 6-5 a 6-6). Dle této analýzy se vybírají další statistické nástroje pro analýzu dat. Hladina významnosti tohoto testu byla zvolena dle zvyklostí analýzy $\alpha=5\%$.

Z analýzy normality dat vyplývá, že hodnota účelové funkce, která závisí na nastavení parametrů křížení a selekce, nemá normální rozdělení (celková p hodnota je menší než 0,05). U modelu FT10 je normalita dosažena dle jednotlivých nastavení pouze v polovině případů. V případě MK10 je normální rozdělení dle nastavení v 7 z 10 případů.

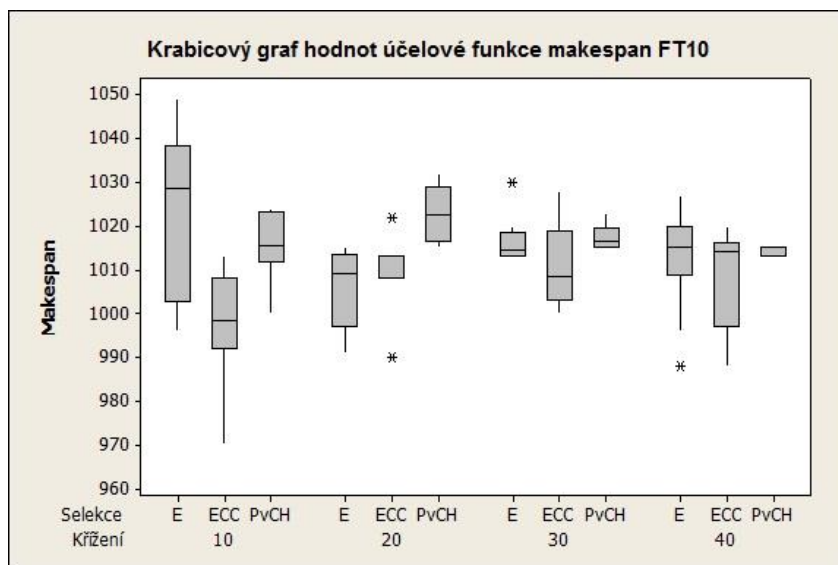


Graf 6-5: Ověření normality dat účelové funkce makespan FT10

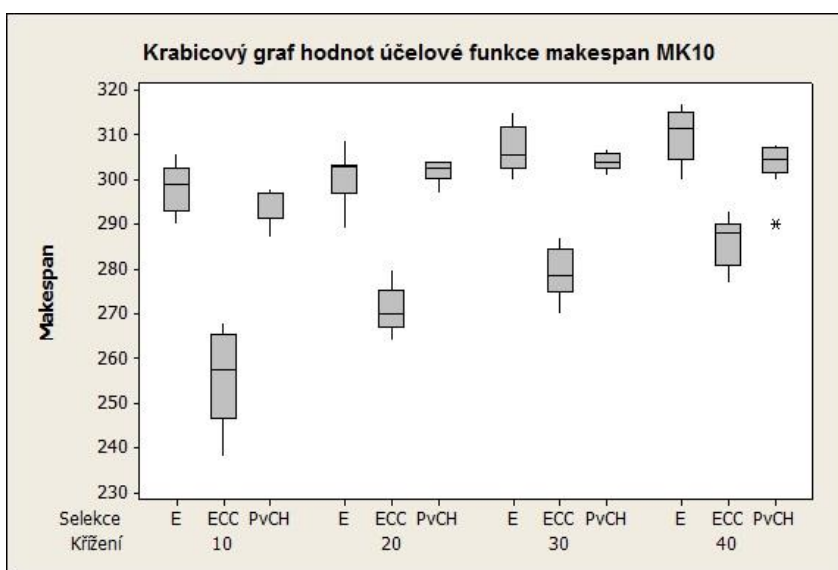


Graf 6-6: Ověření normality dat účelové funkce makespan MK10

Pro následující srovnání nastavení jednotlivých kombinací dle jejich variability a polohy je použit krabicový graf zobrazující minima, první kvartily, mediány, třetí kvartily, maxima a extrémní hodnoty, které jsou vzdáleny od kvartilů na 1,5 násobek rozdílu mezi 1. a 3. kvantilem (označeny hvězdičkou - graf 6-7 a 6-8).



Graf 6-7: Analýza polohy a variability – Krabicový graf funkce makespan modelu FT10



Graf 6-8: Analýza polohy a variability – Krabicový graf funkce makespan modelu MK10

Provedená analýza potvrzuje, že u obou druhů modelů je nejlepším nastavením selekce pomocí ECC a 10% pravděpodobnost křížení. Krabicový graf modelu FT10 nevykazuje zřejmou souvislost mezi jednotlivými nastaveními křížení. U tohoto modelu je však patrná dominance selekce pomocí ECC, až na případ v kombinaci s 20%, kde lze nalézt extrémní maximální i minimální hodnoty funkce makespan. Výsledky analýzy modelu MK10 poukazují na závislost volby obou parametrů. Stejně jako u FT10 je znatelný silný vliv volby selekce (min. hodnoty u ECC), avšak je zde také naznačena závislost hodnoty makespan na křížení, kde při stoupající pravděpodobnosti se zhoršuje hodnota účelové funkce.

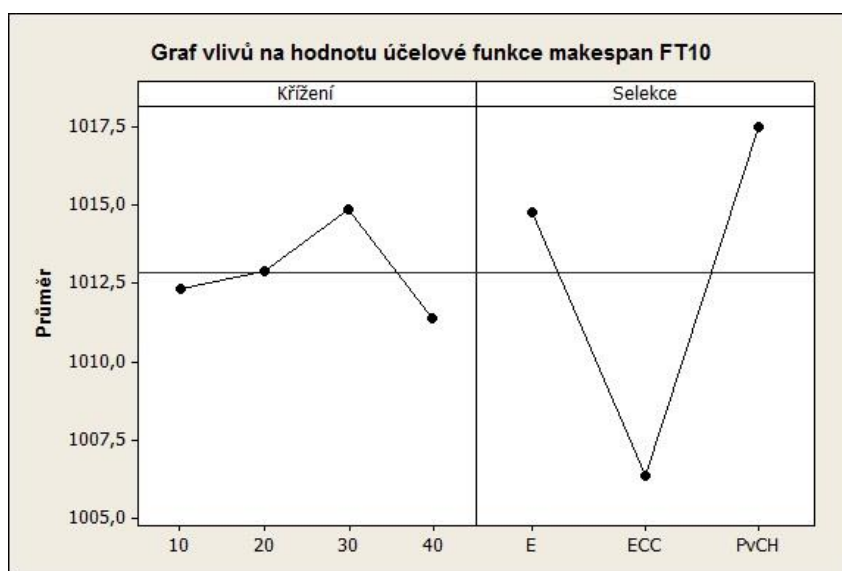
Vzhledem k nenormálnímu rozložení hodnot experimentu nelze analyzovat rovnost průměrů tzn. souvislost mezi jednotlivými nastaveními pomocí testu ANOVA.

Pro porovnání skupin nastavení je použit Kruskal-Wallisův test, kde základní hypotézou je rovnost mediánů.

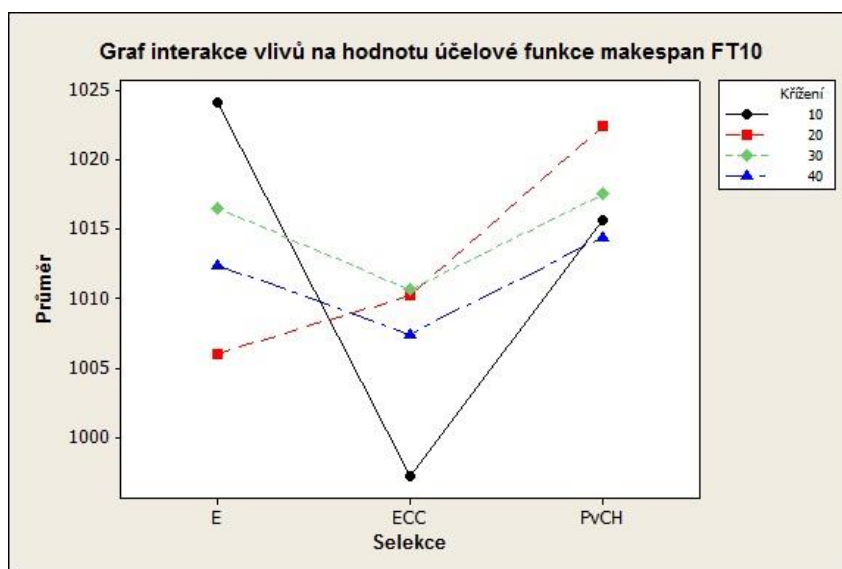
Tento test u modelu FT10 rovnost mediánů pro různé nastavení parametru křížení nezamítl ($p=0,635$) a u selekce zamítl ($p=0$), čímž se potvrzuje silný vliv selekce a slabý vliv nastavení křížení.

Model MK10 vykazuje závislost jak na křížení, tak na selekci (hodnota $p=0$ u obou nastavení). Dominantní kombinací je selekce ECC a 10% pravděpodobnost křížení.

Pro nalezení vhodného nastavení operátorů křížení a selekce je provedena analýza jejich vlivu na účelovou funkci makespan. Tato analýza je doplněna analýzou vzájemných interakcí pro posouzení vzájemného působení výše uvedených operátorů.

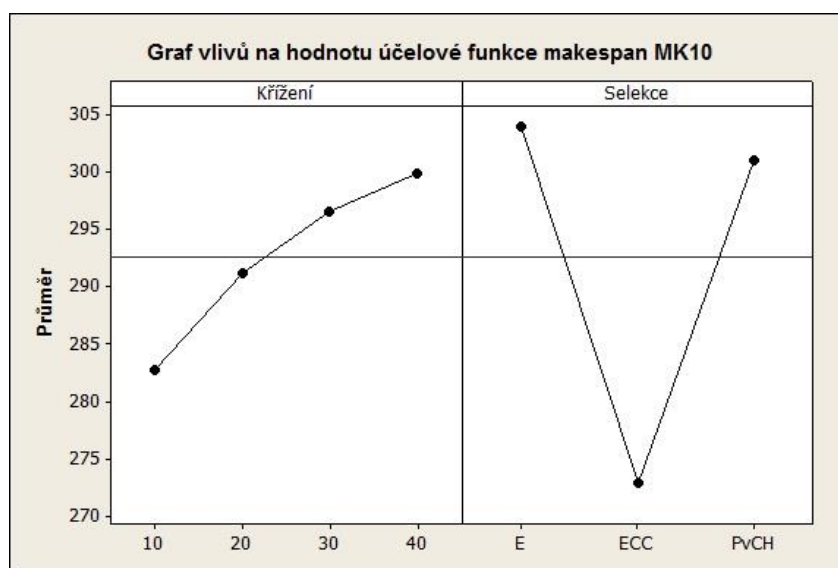


Graf 6-9: Analýza vlivu křížení a selekce na makespan FT10



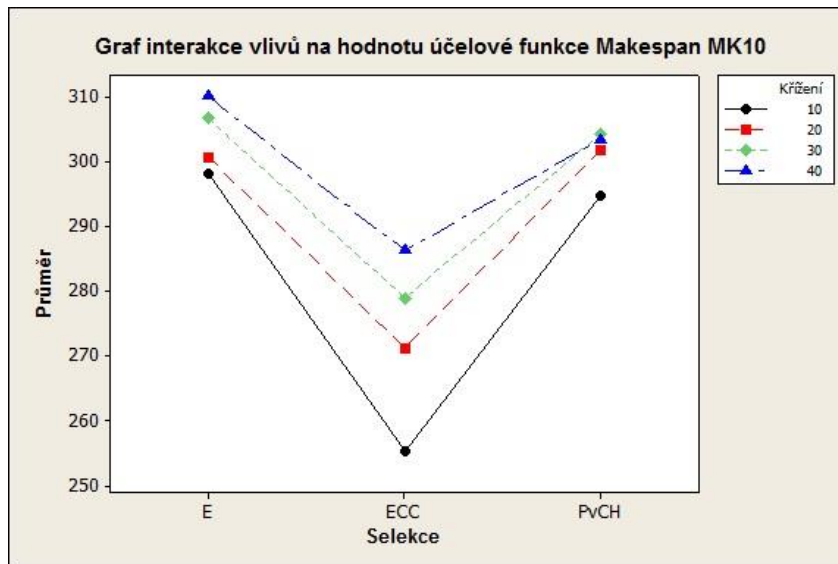
Graf 6-10: Analýza interakce vlivu křížení a selekce na makespan FT10

Z porovnání grafů (model FT10 graf 6-9 a 6-10) vyplývá silná interakce mezi ECC a nastavením křížení na 10%. To vzhledem k tomu, že jako u jediného typu selekce je dosahováno výrazně odlišných výsledků v porovnání s ostatními pravděpodobnostmi křížení (graf 6-10).



Graf 6-11: Analýza vlivu křížení a selekce na makespan MK10

U modelu MK10 je možné pozorovat i vliv křížení, které dosahuje se vzrůstajícím poměrem vyměněných genů horších výsledků hodnoty makespan (graf 6-11).



Graf 6-12: Analýza interakce vlivu křížení a selekce na makespan MK10

Interakce mezi selekcí ECC a 10% pravděpodobností křížení se potvrzuje i na modelu MK10 (graf 6-12), kde je díky většímu výskytu lokálních extrémů tento vliv ještě patrnější.

Z naměřených dat, testů Kruskala-Wallise, analýzy vlivů a jejich vzájemné interakce vyplývá nastavení selekce na ECC a koeficientu křížení na 10% výměny chromosomu jako nejlepší pro dosažení kvalitních (nejbližších optimálních hodnotám)

hodnot funkce makespan. Toto nastavení bude dále použito pro testování hybridizace lokálním prohledáváním.

6.3.7 Volba nastavení mutace hybridizací lokálním prohledáváním

V kapitole 6.3.5 byl navržen princip mutace pomocí hybridizace lokálním prohledáváním. Zásadní otázkou však zůstává míra (kolik operací mutace) použití této metody. Vzhledem k velké časové náročnosti navržené metody je proveden test na modelech FT10 (tab. 6-34) a MK10 (tab. 6-35) při následujících nastaveních:

- 100% (koeficient 1) – toto nastavení je zvoleno pro odhad výkonnosti této metody. Mutace je jediným modifikátorem genu, tzn. jako jediný se zasazuje o vznik nových potomků.
- 50% a (koeficient 0,5) je použit pro odhad náročnosti metody na optimalizační čas a zároveň možnost zlepšení jedinců vytvořených pomocí křížení.
- 20%, 10% a 5% (koeficient 0,1 a 0,05) bývá v častých případech používán pro jednoduchou (bez znalosti o problému např. bez CPA) výměnu genů v rámci mutace [164,165,166].

Tab. 6-34: Výsledky testování FT10 - hybridizace

FT10 - ECC hybridizace (930)						
Koeficient hybridizace	1	0,5	0,2	0,1	0,05	0
Minimum $f(x)$	974	957	977	952	967	970
Maximum $f(x)$	1021	1017	1019	1020	1022	1013
Průměr $f(x)$	993,3	995,8	995,7	994,8	997,9	997,2
Průměrný čas optimalizace [hh:mm:ss]	1:21:28	0:25:28	0:15:39	0:12:18	0:10:42	0:09:02

Z naměřených výsledků modelu FT10 (tab. 6-34) je patrné, že se vzrůstajícím poměrem hybridizace dochází k dosahování o velmi málo lepších výsledků (průměrná hodnota) a navíc v mnohonásobně dlouhém optimalizačním času. Čas optimalizace u koeficientu 1 je více než šestinásobný v porovnání s koeficientem 0,05. Při použití hybridizace lze však nalézt nižší hodnoty makespan, než bez ní.

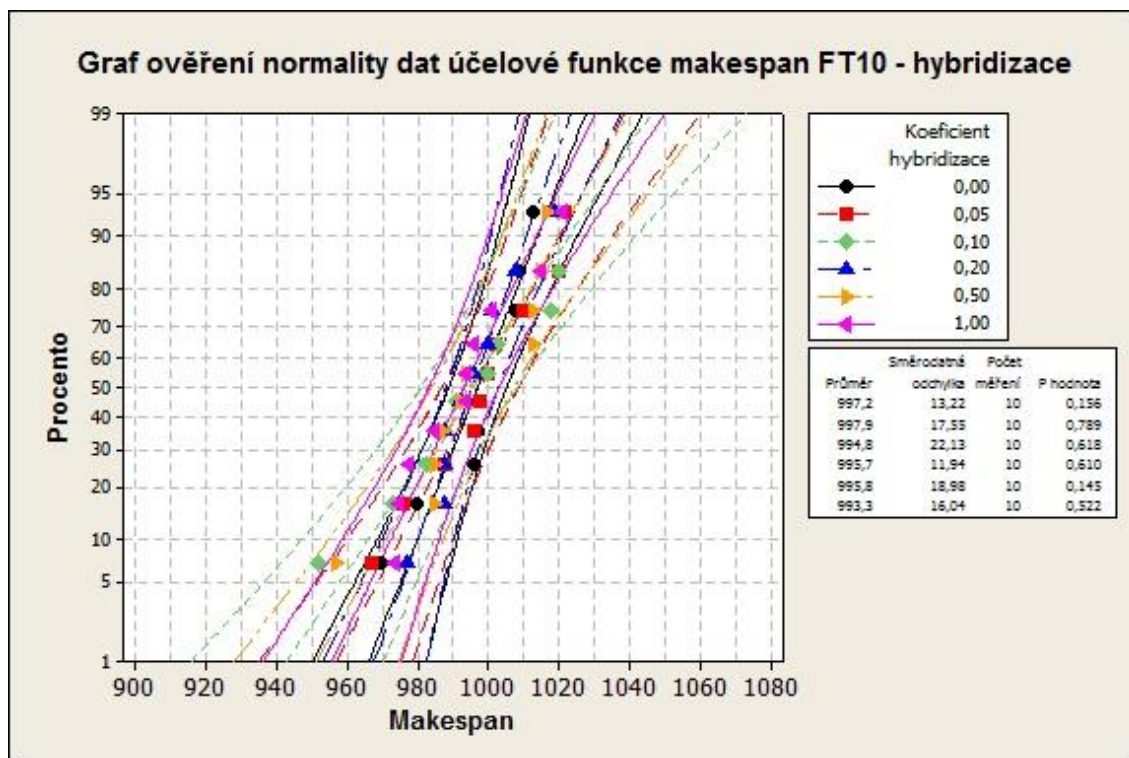
Tab. 6-35: Výsledky testování MK10 - hybridizace

MK10 - ECC hybridizace (165)						
Koeficient hybridizace	1	0,5	0,2	0,1	0,05	0
Minimum $f(x)$	252	248	242	246	242	238
Maximum $f(x)$	268	269	265	264	268	268
Průměr $f(x)$	262,5	261,3	257,2	253,3	252,5	255,3
Průměrný čas optimalizace [hh:mm:ss]	6:33:29	4:07:48	2:41:01	2:10:56	1:57:15	1:50:41

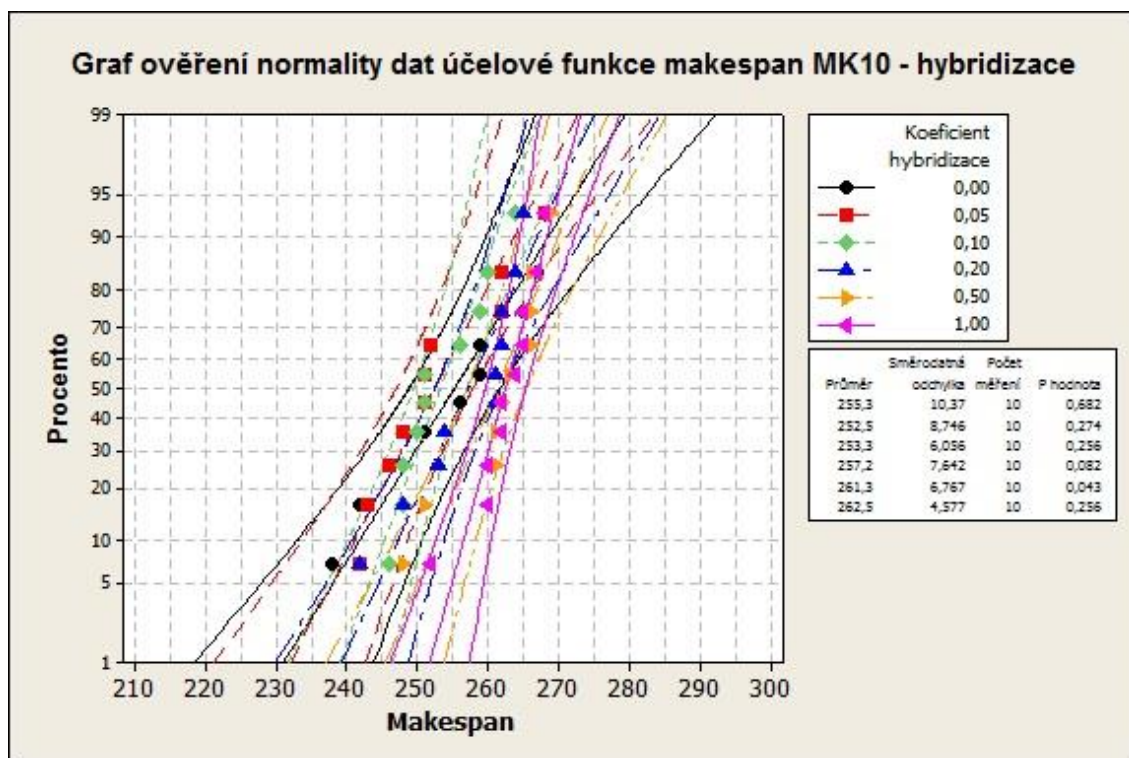
U modelu MK10 nastává situace opačná než u modelu FT10 (tab. 6-35). Čím je poměr hybridizace menší, tím metoda získává lepší výsledky. Na druhou stranu pokud je použita hybridizace, dochází ke snížení celkového průměru hodnoty makespan v 10 experimentech. Testování použití hybridizačního mechanismu na 100% populace ukázalo, že navržený mechanismus, přestože není tak úspěšný jako operace křížení, může částečně optimalizovat rozvrh.

Pro určení nastavení hybridizace vzhledem k statistické odlišnosti jednotlivých nastavení je použit stejný postup jako u stanovení křížení a selekce. Prvním krokem je

test normality obou modelů (FT10 a MK10), pro selekci následujících analýz (graf 6-13 a 6-14).



Graf 6-13: Ověření normality dat účelové funkce makespan FT10



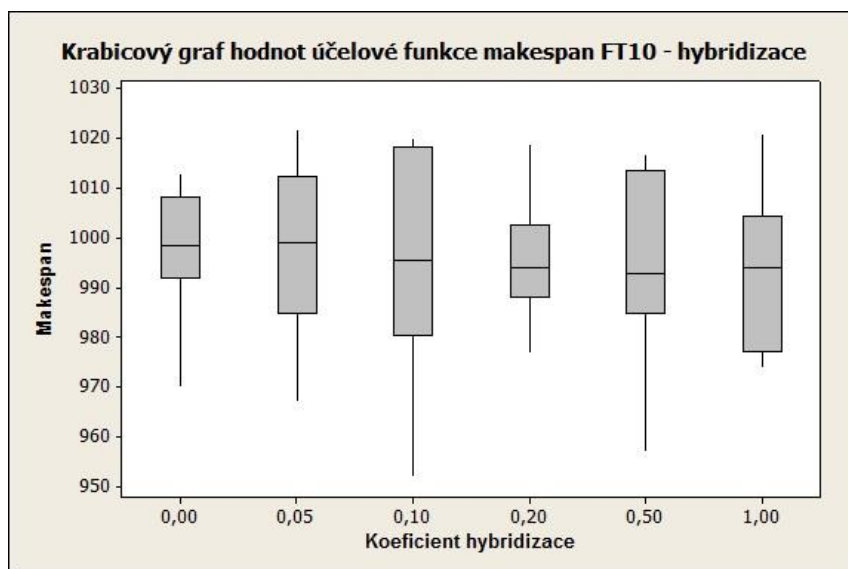
Graf 6-14: Ověření normality dat účelové funkce makespan FT10

Test normality prokázal, že naměřené výsledky modelu FT10 mají normální rozložení, avšak jeden z testovacích kroků u modelu MK10 je nemá (50% hybridizace).

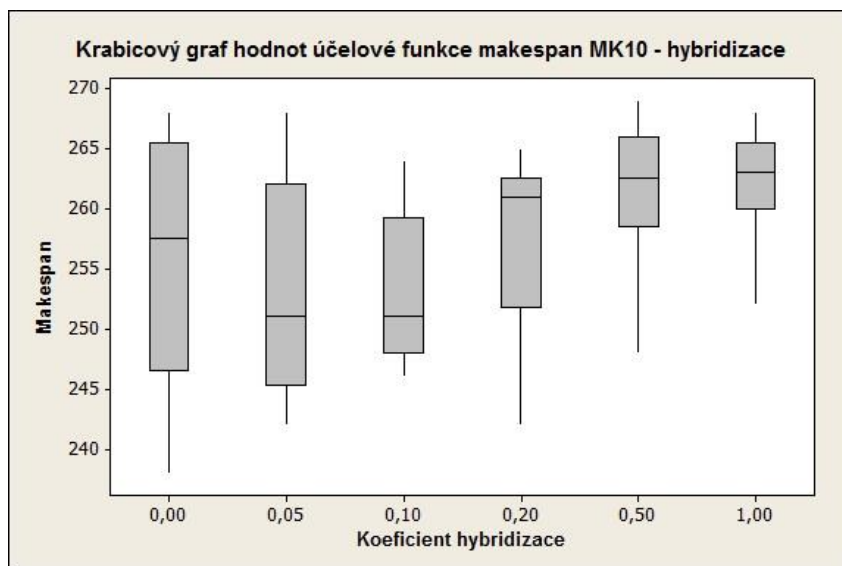
Pro zachování postupu při této analýze, stejně jako u určování vlivu selekce a křížení, je vliv hybridizace posouzen Kruskal-Wallisovým testem, kde základní hypotézou je rovnost mediánů.

V případě modelu FT10 je potvrzena rovnost mediánů ($p=0,9$), tzn. nelze vyloučit, že samotný princip hybridizace nemá statisticky výrazný vliv. U modelu MK10 byla hypotéza o rovnosti mediánů zamítnuta ($p=0$).

Pro posouzení výsledku testu jsou dále porovnány výsledky pomocí krabicových grafů (graf 6-15 a 6-16)



Graf 6-15: Analýza polohy a variability – Krabicový graf makespan modelu FT10-hybridizace



Graf 6-16: Analýza polohy a variability – Krabicový graf makespan modelu MK10-hybridizace

Z analýzy polohy a variability modelu FT10 lze usuzovat, ačkoli se mediány jednotlivých nastavení neliší, že lze dosáhnout kvalitnějších výsledků (bližších optimu při použití hybridizace) s úrovní nastavení parametru hybridizace na 0,1.

Použití hybridizace u modelu MK10 nepřináší vzhledem k nastavenému mechanismu lokálního prohledávání výsledky bližší optimu. Mediány nastavení hybridizace 0,05 a 0,1 jsou však níže, než je tomu u případu bez jejího použití.

Výsledky experimentů ukázaly možný vliv hybridizace. Vzhledem k dosaženým výsledkům účelové funkce makespan (minim, maxim a mediánům) a času optimalizace je pro další etapy testování navrženého algoritmu zvolena úroveň hybridizace na 0,1 (je použita na 10% populace).

6.3.8 Testování zvoleného nastavení na modelech JSSP a FJSSP

Pro testování výkonnosti EA je jako v předchozích případech použito dvou druhů modelů a to JSSP a FJSSP. Je generováno 10 rozvrhů pro každý model a to jak rozvrhů aktivních, tak rozvrhů bez zpoždění.

Parametry EA pak jsou:

- Počet generací = 200
- Velikost populace = 2 x počet operací + 100
- Koeficient křížení = 0,1
- Selektce = ECC
- Koeficient hybridizace = 0,1

Hlavními kritérii hodnocení je jako u předchozích případů (konstruktivní algoritmus CA a algoritmus lokálního prohledávání LS) vzdálenost hodnot účelové funkce od optimální a dále je to procentuální zlepšení výsledku oproti CA a LS.

Důležitým hlediskem aplikace EA je také celkové zlepšení účelové funkce makespan a doby optimalizace.

V tab. 6-36 a 6-37 jsou zobrazeny výsledky optimalizace účelové funkce makespan modelů JSSP generované pomocí aktivních rozvrhů (A) a rozvrhů bez zpoždění ND. Žlutě jsou vyznačeny nejlepší výsledky bez rozdílu druhu rozvrhů A či ND.

Tab. 6-36: Výsledky aktivních rozvrhů účelové funkce makespan JSSP

Aktivní rozvrhy JSSP							
Model	Optimum	Min.	Odchylka od optima	Max.	Odchylka od optima	Průměr	Odchylka od optima
FT06	55	55	0,00%	55	0,00%	55	0,00%
FT10	930	952	2,37%	1020	9,68%	994,8	6,97%
FT20	1165	1208	3,69%	1293	10,99%	1260,2	8,17%
SW11	2983	3516	17,87%	3616	21,22%	3578,3	19,96%
LA2	655	655	0,00%	667	1,83%	657,7	0,41%
LA19	842	848	0,71%	885	5,11%	860,4	2,19%
LA21	1046	1122	7,27%	1192	13,96%	1164,7	11,35%
LA27	1235	1350	9,31%	1407	13,93%	1378,7	11,64%
LA30	1355	1382	1,99%	1480	9,23%	1431,9	5,68%
LA40	1222	1298	6,22%	1348	10,31%	1314,3	7,55%
Průměrná odchylka			4,94%		9,62%		7,39%

V případě A je pomocí EA dosaženo optima či hodnotě blízké optimu ve třech případech (FT06, LA2 a LA 19). U nejjednoduššího modelu FT06 je optimum dosaženo ve všech 10-ti opakováních. V ostatních případech, s výjimkou modelu SW11, jehož nejlepší výsledek byl vzdálen cca 20% od optima, je odchylka od optima přibližně 10%. Rozdíly mezi minimální a maximální dosaženou hodnotou jsou s ohledem na praktické aplikace relativně malé a činí maximálně 4%, vázáno na optimální hodnotu makespan.

Tab. 6-37: Výsledky rozvrhů bez zpoždění účelové funkce makespan JSSP

Rozvrhy bez zpoždění JSSP							
Model	Optimum	Min.	Odchylka od optima	Max.	Odchylka od optima	Průměr	Odchylka od optima
FT06	55	57	3,64%	58	5,45%	57,2	4,00%
FT10	930	960	3,23%	988	6,24%	976,7	5,02%
FT20	1165	1201	3,09%	1252	7,47%	1221,5	4,85%
SW11	2983	3402	14,05%	3524	18,14%	3490	17,00%
LA2	655	672	2,60%	681	3,97%	675,4	3,11%
LA19	842	875	3,92%	875	3,92%	875	3,92%
LA21	1046	1090	4,21%	1129	7,93%	1109,8	6,10%
LA27	1235	1310	6,07%	1340	8,50%	1327,6	7,50%
LA30	1355	1355	0,00%	1383	2,07%	1370,6	1,15%
LA40	1222	1286	5,24%	1304	6,71%	1296,6	6,10%
Průměrná odchylka			4,60%		7,04%		5,88%

U rozvrhů bez zpoždění bylo dosaženo pouze u jednoho z modelů optima, což je způsobeno samotnou podstatou rozvrhů bez zpoždění. Na druhou stranu dosáhly ND v 6 z 10 modelů lepší makespan než aktivní rozvrhy. V průměru ND dosáhly menší odchylky od optima než A.

Tak jako u modelů JSSP je testována odchylka od optima i u modelů FJSSP v generaci aktivních rozvrhů (A) a rozvrhů bez zpoždění (ND). V mnohých případech dosáhly oba dva stejných výsledků. Nejlepší výsledky z obou jsou pak vyznačeny žlutě (tab. 6-38 a 6-39).

Tab. 6-38: Výsledky aktivních rozvrhů účelové funkce makespan FJSSP

Aktivní rozvrhy FJSSP							
Model	Optimum	Min.	Odchylka od optima	Max.	Odchylka od optima	Průměr	Odchylka od optima
MK01	36	42	16,67%	42	16,67%	42	16,67%
MK02	24	32	33,33%	35	45,83%	33,2	38,33%
MK03	204	204	0,00%	204	0,00%	204	0,00%
MK04	48	66	37,50%	68	41,67%	67	39,58%
MK05	168	177	5,36%	187	11,31%	182	8,33%
MK06	33	74	124,24%	82	148,48%	77,3	134,24%
MK07	133	157	18,05%	171	28,57%	165,9	24,74%
MK08	523	523	0,00%	523	0,00%	523	0,00%
MK09	299	320	7,02%	339	13,38%	332,2	11,10%
MK10	165	246	49,09%	264	60,00%	253,3	53,52%
Průměrná odchylka			29,13%		36,59%		32,65%

Tab. 6-39: Výsledky rozvrhů bez zpoždění účelové funkce makespan FJSSP

Rozvrhy bez zpoždění FJSSP							
Model	Optimum	Min.	Odchylka od optima	Max.	Odchylka od optima	Průměr	Odchylka od optima
MK01	36	42	16,67%	42	16,67%	42	16,67%
MK02	24	31	29,17%	34	41,67%	32,3	34,58%
MK03	204	204	0,00%	204	0,00%	204	0,00%
MK04	48	66	37,50%	67	39,58%	66,8	39,17%
MK05	168	176	4,76%	179	6,55%	177,8	5,83%
MK06	33	79	139,39%	85	157,58%	81,8	147,88%
MK07	133	162	21,80%	170	27,82%	167,3	25,79%
MK08	523	523	0,00%	523	0,00%	523	0,00%
MK09	299	314	5,02%	322	7,69%	317,8	6,29%
MK10	165	256	55,15%	264	60,00%	259,6	57,33%
Průměrná odchylka			30,95%		35,76%		33,35%

Odchylka od optima je v případě modelů FJSSP větší než u modelů JSSP. To je dáno větší kombinatorickou složitostí. Na druhou stranu u složitějších modelů MK03 a MK08 našel EA optima jak v případě A i ND a to ve všech 10 případech opakování experimentu.

Dalším kritériem hodnocení EA je schopnost nalézt lepší řešení (nižší hodnota makespan) než metoda konstruktivního algoritmu a lokálního prohledávání. To je měřeno zlepšením makespanu oproti zmíněným metodám, kde základem (100%) je hodnota optima.

Porovnávány jsou pak nejlepší hodnoty dosažené v rámci generování A a ND rozvrhů pomocí CA a LS mezi příslušnými modely JSSP (tab. 6-40 a 6-41) a FJSSP (tab. 6-42 a 6-43).

Tab. 6-40: Zlepšení aktivních rozvrhů účelové funkce makespan JSSP pomocí EA

Aktivní rozvrhy JSSP						
Model	Optimum	Nejlepší CA	Nejlepší LS	Nejlepší EA	Zlepšení vůči CA	Zlepšení vůči LS
FT06	55	65	55	55	18,18%	0,00%
FT10	930	1178	1052	952	24,30%	10,75%
FT20	1165	1539	1501	1208	28,41%	25,15%
SW11	2983	4045	3924	3516	17,73%	13,68%
LA2	655	830	733	655	26,72%	11,91%
LA19	842	1120	966	848	32,30%	14,01%
LA21	1046	1265	1209	1122	13,67%	8,32%
LA27	1235	1567	1545	1350	17,57%	15,79%
LA30	1355	1565	1523	1382	13,51%	10,41%
LA40	1222	1531	1380	1298	19,07%	6,71%
Průměrné zlepšení					21,15%	11,67%

Tab. 6-41: Zlepšení rozvrhů bez zpoždění účelové funkce makespan JSSP pomocí EA.

Rozvrhy bez zpoždění JSSP						
Model	Optimum	Nejlepší CA	Nejlepší LS	Nejlepší EA	Zlepšení vůči CA	Zlepšení vůči LS
FT06	55	61	60	57	7,27%	5,45%
FT10	930	1074	1074	960	12,26%	12,26%
FT20	1165	1267	1267	1201	5,67%	5,67%
SW11	2983	3530	3530	3402	4,29%	4,29%
LA2	655	756	756	672	12,82%	12,82%
LA19	842	940	940	875	7,72%	7,72%
LA21	1046	1264	1264	1090	16,63%	16,63%
LA27	1235	1442	1442	1310	10,69%	10,69%
LA30	1355	1534	1534	1355	13,21%	13,21%
LA40	1222	1434	1394	1286	12,11%	8,84%
Průměrné zlepšení					10,27%	9,76%

EA dosáhlo průměrně oproti CA o cca 20% lepších výsledků u A rozvrhů (tab. 6-40), v případě ND pak o 10% (tab. 6-41). Větší úspěšnost je způsobena zejména neschopností ND dosáhnout optima a také tím, že rozvrhy generované pomocí CA u A rozvrhů nejsou tak kvalitní jako u ND.

Navržený algoritmus EA dosáhl o cca 10% lepších výsledků než LS. To je způsobeno zejména pokročilejší technikou u EA, která uvízla v daleko silnějších extrémech než LS, které často nebylo schopno zlepšit výsledek CA.

Lepších výsledků než u LS bylo dosaženo i u modelů FJSSP (tab. 6-42 a 6-43).

Tab. 6-42: Zlepšení aktivních rozvrhů účelové funkce makespan FJSSP pomocí EA.

Aktivní rozvrhy FJSSP						
Model	Optimum	Nejlepší CA	Nejlepší LS	Nejlepší EA	Zlepšení vůči CA	Zlepšení vůči LS
MK01	36	51	45	42	25,00%	8,33%
MK02	24	38	38	32	25,00%	25,00%
MK03	204	204	204	204	0,00%	0,00%
MK04	48	80	79	66	29,17%	27,08%
MK05	168	198	198	177	12,50%	12,50%
MK06	33	94	94	74	60,61%	60,61%
MK07	133	208	208	157	38,35%	38,35%
MK08	523	524	523	523	0,19%	0,00%
MK09	299	350	350	320	10,03%	10,03%
MK10	165	287	287	246	24,85%	24,85%
Průměrné zlepšení					22,57%	20,68%

Tab. 6-43: Zlepšení rozvrhů bez zpoždění účelové funkce makespan FJSSP pomocí EA.

Rozvrhy bez zpoždění FJSSP						
Model	Optimum	Nejlepší CA	Nejlepší LS	Nejlepší EA	Zlepšení vůči CA	Zlepšení vůči LS
MK01	36	47	46	42	13,89%	11,11%
MK02	24	44	36	31	54,17%	20,83%
MK03	204	204	204	204	0,00%	0,00%
MK04	48	77	77	66	22,92%	22,92%
MK05	168	189	189	176	7,74%	7,74%
MK06	33	93	93	79	42,42%	42,42%
MK07	133	200	200	162	28,57%	28,57%
MK08	523	523	523	523	0,00%	0,00%
MK09	299	349	349	314	11,71%	11,71%
MK10	165	266	266	256	6,06%	6,06%
Průměrné zlepšení					18,75%	15,14%

Průměrné zlepšení u FJSSP v případě rozvrhů A dosáhlo úrovně cca 20%, u ND pak 15%-18% a to i přes to, že u dvou modelů (MK03 a MK08) nebylo nalezeno lepší řešení než u CA nebo LS. To je způsobeno tím, že všechny metody byly schopné u těchto modelů nalézt optimum.

Posledním důležitým kritériem je pak časová náročnost optimalizace pomocí EA a tedy celkový přínos oproti použití předcházejících metod CA a LS. Pro srovnání jednotlivých metod jsou použity ve všech případech (CA, LS a EA) nejlepší dosažené výsledky (tab. 6-44 a 6-45). V této analýze jsou hodnoty účelové funkce makespan převedeny na minuty (jedna časová jednotka = jedna minuta). Při porovnání časových výsledků se vychází z toho, že optimalizace probíhá v rámci procesu výroby, tzn. je přínos považován za takovou situaci, kde rozdíl mezi výsledkem generovaným pomocí CA a GA je větší než čas potřebný pro optimalizaci.

Zlepšení tedy udává rozdíl mezi výsledkem metody CA a EA s tím, že celkový přínos je zlepšení ponížené o nutnou časovou investici do metody EA.

Tab. 6-44: Porovnání časového přínosu optimalizace EA - JSSP

Model	Nejlepší řešení CA [hod:min:sek]	Čas CA [hod:min:sek]	Nejlepší řešení EA [hod:min:sek]	Čas EA [hod:min:sek]	Zlepšení [hod:min:sek]	Přínos [hod:min:sek]
FT06	1:01:00	0:00:01	0:55:00	0:01:53	0:06:00	0:04:07
FT10	17:54:00	0:00:01	15:52:00	0:12:18	2:02:00	1:49:42
FT20	21:07:00	0:00:03	20:01:00	0:15:10	1:06:00	0:50:50
SW11	58:50:00	0:00:03	52:42:00	8:25:00	6:08:00	-2:17:00
LA2	12:36:00	0:00:03	10:55:00	0:03:42	1:41:00	1:37:18
LA19	15:40:00	0:00:05	14:08:00	0:11:45	1:32:00	1:20:15
LA21	21:04:00	0:00:06	18:10:00	0:28:08	2:54:00	2:25:52
LA27	24:02:00	0:00:06	21:50:00	0:55:14	2:12:00	1:16:46
LA30	25:34:00	0:00:07	22:35:00	0:54:31	2:59:00	2:04:29
LA40	23:54:00	0:00:21	21:26:00	1:00:18	2:28:00	1:27:42

Výsledky optimalizace JSSP modelů ukazují, že ve všech případech (vyjma modelu SW11) lze pomocí EA dosáhnout znatelných časových úspor. EA u modelu SW11 dosáhla více než 6 hodinového zlepšení, avšak čas optimalizace byl větší než 8 hodin. V takovém případě lze při znalosti modelu doporučit optimalizovat pořadí zakázek v předstihu.

Tab. 6-45: Porovnání časového přínosu optimalizace EA - FJSSP

Model	Nejlepší řešení CA [hod:min:sek]	Čas CA [hod:min:sek]	Nejlepší řešení EA [hod:min:sek]	Čas EA [hod:min:sek]	Zlepšení [hod:min:sek]	Přínos [hod:min:sek]
MK01	0:47:00	0:00:02	0:42:00	0:05:01	0:05:00	- 0:00:01
MK02	0:38:00	0:00:02	0:31:00	0:07:38	0:07:00	- 0:00:38
MK03	3:24:00	0:00:06	3:24:00	0:37:58	0:00:00	- 0:37:58
MK04	1:17:00	0:00:03	1:06:00	0:16:32	0:11:00	- 0:05:32
MK05	3:09:00	0:00:03	2:54:00	0:13:21	0:15:00	0:01:39
MK06	1:33:00	0:00:06	1:14:00	0:40:15	0:19:00	- 0:21:15
MK07	3:20:00	0:00:04	2:37:00	0:25:05	0:43:00	0:17:55
MK08	8:43:00	0:00:08	8:43:00	1:08:17	0:00:00	- 1:08:17
MK09	5:49:00	0:00:09	5:14:00	1:34:28	0:35:00	- 0:59:28
MK10	4:26:00	0:00:11	4:06:00	2:10:56	0:20:00	- 1:50:56

Výsledky časového přínosu optimalizace pomocí EA ukazují, že u testovaných modelů lze použít optimalizaci pouze v případech, kdy je dostatek času v období přípravy výroby na další plánovací horizont (tab. 6-45 žlutě). V některých případech došlo k malým (MK01, MK02, MK04 a MK05 zlepšení cca 5-10 minut) či žádným zlepšením (MK03 a MK08). U některých modelů (MK06, MK07, MK09 a MK10) však zlepšení dosahovalo cca 20-40 minut, což může mít pro praktické aplikace vliv.

Z testování EA vyplývá, že lze dosáhnout časových úspor, které se však případ od případu liší. Jak se na ilustračním porovnání přínosu metody EA ukázalo, je zásadní, kdy bude optimalizace probíhat. Za chodu výroby by měla být redukce makespanu větší, než je délka optimalizace. V případě optimalizace před náběhem výroby do výrobního systému mohou být přínosy daleko větší, je však třeba znát délku optimalizace. Tu lze však bez předchozího testování modelů jen obtížně odhadnout.

Proto jedním z požadavků na optimalizační nástroj je omezení optimalizačního času definované plánovačem. Tento požadavek je dále rozpracován v následující kapitole zabývající se tzv. „automatickou optimalizací“.

6.4 Automatická optimalizace

Funkce automatická optimalizace vychází z běžných potřeb dílenského plánovače, u kterého se nepředpokládá znalost pokročilých heuristických algoritmů a zároveň má nedostatek času pro to, aby se snažil najít vhodné nastavení parametrů těchto algoritmů vzhledem k rozvrhovanému modelu.

Hlavním cílem dílenského plánovače je poskytnout výrobě v určený čas reálný rozvrh, který by co nejvíce splňoval požadavky dílny a zároveň požadavky kladené zákazníkem. Jak již bylo popsáno, tyto požadavky mohou mít často protichůdný charakter. Klasickým příkladem je snaha dílenského plánovače co nejdříve odvést plánované úlohy (makespan), na stranu druhou je tlačěn k plnění zákaznických termínů (tardiness) a zároveň i k vysokému využití zdrojů.

Proto v následně popsané metodě dílenský plánovač pouze zadává čas, kdy bude potřeba finální rozvrh výroby (za jak dlouho, či přesný čas dokončení optimalizace) a účelovou funkci, dle které má být hodnocena optimalizace.

Vzhledem k výše popsaným požadavkům obsahuje automatická optimalizace konstruktivní algoritmus (CA), který přes to, že nedokáže poskytnout výsledky blízké optimu, má z analyzovaných metod nejnižší časové požadavky na poskytnutí jakéhokoli rozvrhu.

Druhou použitou metodou je pak lokální prohledávání (LS) jednoduchým prohozením, které dokáže v některých případech dosáhnout lepšího výsledku než CA, avšak často uvízne v lokálním extrému a optimalizace trvá déle, než je navržení řešení pomocí CA.

Třetí metodou aplikovanou v rámci automatické optimalizace je testovaný evoluční algoritmus (EA), který díky svým pokročilým technikám optimalizace, založených na evolučních strategiích a hybridizaci, dokáže nalézt řešení blízká optimu, avšak za delší optimalizační čas než CA a LS.

V rámci automatické optimalizace se tedy hierarchicky použije nejprve CA, které pomocí pravidla nejkratší procesní čas (SPT) odhadne dobu tvorby rozvrhu, aby

následně použil pravidla nejdelší procesní čas (LPT), nejvíce práce zbývá (MWKR), první příchozí je prvním zpracovaným (FIFO) a 10 rozvrhů náhodným výběrem (RS).

Odhad doby trvání SPT je dále použit pro povolení či zamítnutí tvorby rozvrhu pomocí LPT. Následující pravidla jsou použita na základě zbývajících času a odhadu trvání optimalizace vycházejících z průměrné doby tvorby předchozích rozvrhů. Tento princip kompenzuje rozdílné využití PC prostředků v čase.

Pokud zbývá čas na další optimalizaci je, vzhledem k riziku uvíznutí algoritmu LS v lokálním extrému, použito pro lokální prohledávání jako počáteční řešení pouze to s nejlepší hodnotou účelové funkce. Algoritmus pak provede odhad trvání výpočtu jedné sady prohození na základě trvání CA s tím, že předpokládá, že všechna prohození vygenerují reálné rozvrhy. Odhad času následující iterace prohození předpovídá z trvání prvního prohození. Stejně jako v případě CA je odhad budoucí optimalizace určen průměry předchozích iterací. Optimalizace probíhá, dokud je dostupný optimalizační čas či LS neuvízlo v lokálním extrému. Pokud LS uvízne v některém z lokálních extrémů a je dostupný čas pro optimalizaci, je spuštěn EA.

Největším problémem při aplikaci navrženého EA je odhad času, který je potřeba pro optimalizaci. Generování rozvrhů není závislé pouze na parametrech nastavení EA, ale i na kombinatorické složitosti. Zásadní vliv na dobu optimalizace (viz kapitola 6.3.6) má operátor velikost populace, který určuje počet rozvrhů v jedné generaci, a operátor počet generací, který určuje počet iterací EA.

Nalezení správného nastavení těchto parametrů vzhledem k celkovému zbývajícím času optimalizace by vyžadovalo časově náročné testování při každé změně, ať již v modelu či v požadavku na dostupný čas.

Místo použití fixní velikosti populace je v EA aplikován v automatické optimalizaci použit jeden z principů adaptivních algoritmů manipulujících s velikostí populace (APSS) [167, 168, 169, 170].

Většina z těchto adaptivních postupů zavádí místo operátorů velikost populace a počet generací řadu jiných operátorů pro zajištění řízení dvou zmiňovaných. Principy těchto adaptivních postupů jsou v rozporu s požadovanou jednoduchostí EA. Přesto je pro řízení počtu generací a velikosti populace využit adaptivní princip PROFIGA (Population Resizing on Fitness Improvement GA) [169], který lze modifikovat na řízení zmiňovaných operátorů v závislosti na dostupném času optimalizace. PROFIGA manipuluje s velikostí populace (zmenšuje a zvětšuje) na základě hodnoty účelové funkce u nejlepšího z jedinců. Populace roste o faktor X v případě, že bylo nalezeno lepší řešení. Pokud se populace nezlepšuje po určitou dobu (doporučené nastavení je 5 generací), tak je populace zmenšena o faktor 0,1 (populace se zmenší o 10%). Pokud nenastane ani jedna z možností, velikost populace se nemění. Počáteční velikost populace je stanovena na 10 jedinců. Faktor X (20) pak je:

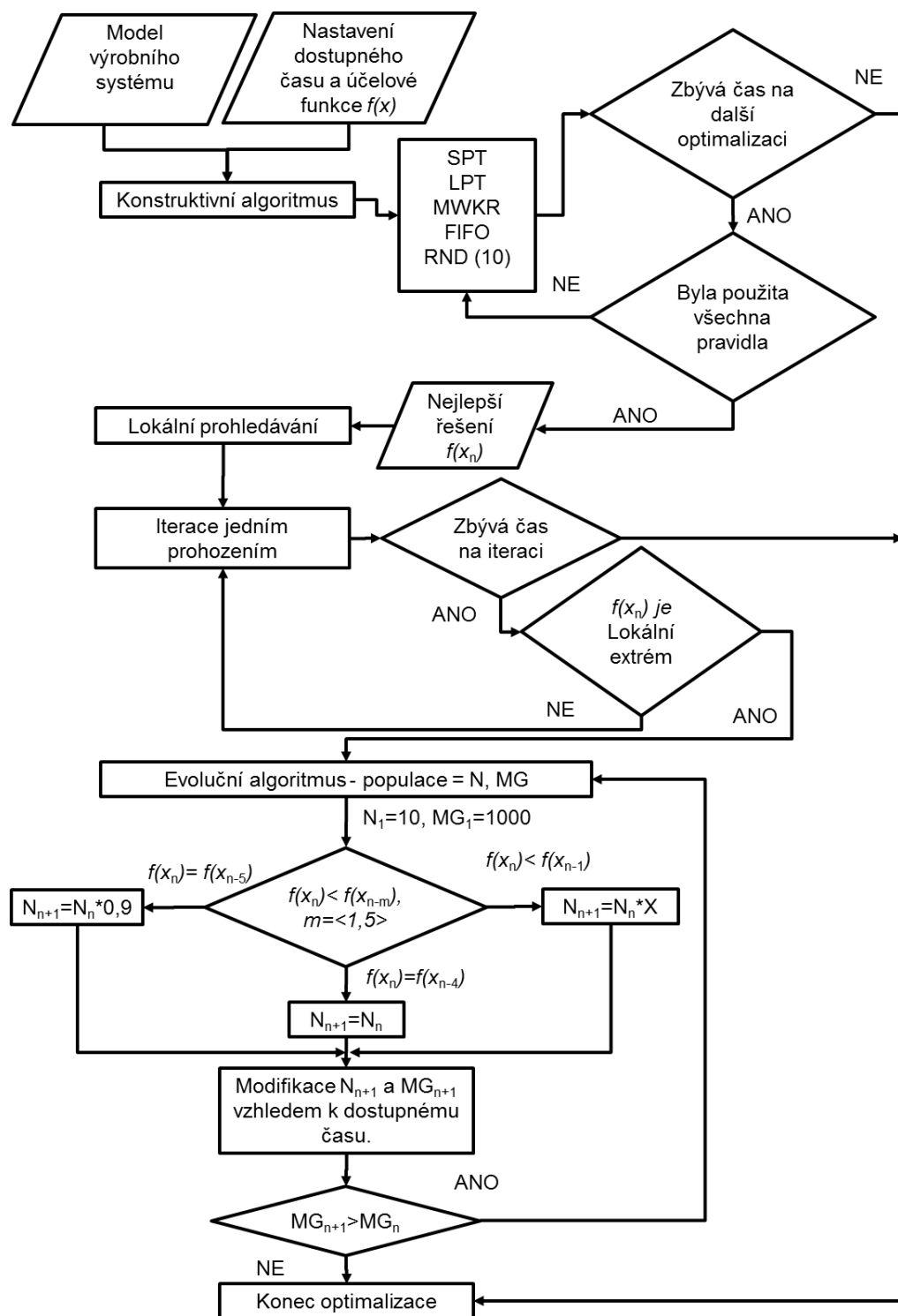
- FR –faktor růstu (0,5).
- MG –maximální počet generací je fixně určen. Při této aplikaci je MG nastaveno na 1000 s tím, že v průběhu optimalizace a ubývajícím dostupného času se tato hodnota mění na základě predikce o délce trvání minulých iterací a nastavení velikosti populace pro příští generaci. Maximální počet generací tedy může klesat či stoupat v průběhu optimalizace.
- SG – pořadové číslo současné generace.

- $\max FIT_n$ – nejlepší současná hodnota účelové funkce.
- $\max FIT_p$ – předchozí nejlepší hodnota účelové funkce.
- $\text{inimax} FIT$ – nejlepší hodnota účelové funkce u první generace.

$$X = FR * (MG - SG) * \frac{\max FIT_n - \max FIT_p}{\text{inimax} FIT} \quad (20)$$

Při zvětšování velikosti populace se doporučuje doplnit současná řešení o nově náhodně vygenerovaná. Při aplikaci principu P_{Ro}FIGA je využit princip kontroly klonů při selekci ECC (viz kapitola 6.3.3), kdy řešení, která nejsou jedinečná, jsou nahrazena odlišnými jedinci, kteří nebyli úspěšnými v rámci selekce. Teprve v případě, že nejsou dostupná žádná unikátní řešení, je populace doplněna náhodně generovanými jedinci. Velikost navržené populace je také ovlivněna zbývajícím časem pro optimalizaci. Pokud by byla následná generace tak velká, že by byl překročen dostupný čas optimalizace, pak je odpovídajícím způsobem snížen počet jedinců pro následující generaci a maximální počet generací.

Celkový průběh automatické optimalizace je pak shrnut na obrázku 6-11.



Obr. 6-11: Postup tvorby rozvrhu pomocí automatické optimalizace

Takto navržený algoritmus je testován jako v případě EA na 10 modelech JSSP a 10 modelech FJSP s nastavením parametrů stanovených v kapitolách 6.3.6 až 6.3.8:

- Doba optimalizace: 15 minut (vychází ze zkušeností z realizace projektů řízení výroby a četných konzultací s odborníky na plánování a řízení výroby strojírenských podniků).

- Konstruktivní algoritmus (CA):
 - Prioritní pravidla: SPT, LPT, MWKR, FIFO a 10 rozvrhů pomocí RS.
 - Druhy rozvrhů: Aktivní a bez zpoždění.
- Lokální prohledávání (LS):
 - Počáteční řešení: Řešení dané z CA s nejlepší hodnotou účelové funkce makespan.
 - Druh rozvrhu: Dle řešení daného z CA s nejlepší hodnotou účelové funkce makespan.
- Evoluční algoritmus:
 - Velikost populace: Automaticky dle dostupného času optimalizace a vývoje účelové funkce makespan.
 - Počet generací: Automaticky dle dostupného času optimalizace a vývoje účelové funkce makespan.
 - Koeficient křížení: 0,1 (viz kapitola 6.3.6).
 - Selektce: ECC (viz kapitola 6.3.6).
 - Koeficient hybridizace: 0,1 (viz kapitola 6.3.7).
 - Druhy rozvrhů: Aktivní a bez zpoždění.

Pro porovnání klasického přístupu EA a EA uvnitř automatické optimalizace (AO) jsou posuzovány kritérii vzdálenost od optima (vztaheno k nejlepšímu řešení), minimální, maximální a průměrné hodnoty účelové funkce makespan (tab. 6-46 až 6-49). Každý z modelů dle uvedeného nastavení je testován v rámci 10-ti experimentů. Výsledky LS a CA včetně časové náročnosti jsou v příslušných kapitolách (6.1 a 6.2).

Tab. 6-46: Porovnání EA a AO modelů JSSP – aktivní rozvrhy

Model	Optimum	Odchylka od optima EA	Odchylka od optima AO	Min EA	Min AO	Max EA	Max AO	Průměr EA	Průměr AO	S.odchylka EA	S.odchylka AO
FT06	55	0,00%	0,00%	55	55	55	55	55	55	0,00	0,00
FT10	930	2,37%	4,84%	952	975	1020	998	994,8	987,1	21,00	7,12
FT20	1165	3,69%	2,15%	1208	1190	1293	1305	1260	1264,3	21,66	31,69
SW11	2983	17,87%	20,99%	3516	3609	3616	3800	3578	3712,7	37,79	54,08
LA2	655	0,00%	0,00%	655	655	667	671	657,7	658,3	3,82	5,62
LA19	842	0,71%	0,71%	848	848	885	872	860,4	860	11,93	6,53
LA21	1046	7,27%	8,60%	1122	1136	1192	1191	1165	1164,8	17,80	18,41
LA27	1235	9,31%	11,01%	1350	1371	1407	1455	1379	1413,6	15,89	25,89
LA30	1355	1,99%	6,42%	1382	1442	1480	1501	1432	1483,5	27,45	20,67
LA40	1222	6,22%	7,45%	1298	1313	1348	1386	1314	1348,4	14,38	24,30
Průměrná vzdálenost od optima		4,94%	6,22%					Průměrná S.odchylka		17,17	19,43

Tab. 6-47: Porovnání EA a AO modelů JSSP – rozvrhy bez zpoždění

Model	Optimum	Odchylka od optima EA	Odchylka od optima AO	Min EA	Min AO	Max EA	Max AO	Průměr EA	Průměr AO	S.odchylka EA	S.odchylka AO
FT06	55	3,64%	3,64%	57	57	58	57	57,2	57	0,40	0,00
FT10	930	3,23%	3,33%	960	961	988	1015	976,7	988,6	11,43	12,37
FT20	1165	3,09%	4,21%	1201	1214	1252	1257	1222	1237,3	17,10	14,68
SW11	2983	14,05%	15,89%	3402	3457	3524	3634	3490	3546,4	39,86	48,76
LA2	655	2,60%	1,98%	672	668	681	681	675,4	673,9	3,50	3,73
LA19	842	3,92%	3,92%	875	875	875	875	875	875	0,00	0,00
LA21	1046	4,21%	5,35%	1090	1102	1129	1150	1110	1133,2	14,40	13,56
LA27	1235	6,07%	7,77%	1310	1331	1340	1393	1328	1348,8	9,50	17,79
LA30	1355	0,00%	0,89%	1355	1367	1383	1431	1371	1401,4	6,95	20,53
LA40	1222	5,24%	5,97%	1286	1295	1304	1316	1297	1308,7	5,92	5,85
Průměrná vzdálenost od optima		4,60%	5,30%					Průměrná S.odchylka		10,90	13,73

Z výsledků testovaných modelů JSSP vyplývá, že EA nachází kvalitnější výsledky (nižší hodnoty účelové funkce) než AO, avšak rozdíly mezi výsledky jsou minimální. Rozdíly jsou způsobené zejména rozdílnou délkou optimalizace, kdy u modelů LA21-40 a SW11 je do optimalizace investován znatelně kratší čas (viz tab. 6-44. kapitola 6.3.8). Porovnáním aktivních rozvrhů a rozvrhů bez zpoždění lze konstatovat, že rozvrhy bez zpoždění poskytují v průměru výsledky blíže optima, ačkoli aktivní rozvrhy byly schopné nalézt optimum častěji.

Tab. 6-48: Porovnání EA a AO modelů FJSSP – aktivní rozvrhy

Model	Optimum	Odchylka od optima EA	Odchylka od optima AO	Min EA	Min AO	Max EA	Max AO	Průměr EA	Průměr AO	S.odchylka EA	S.odchylka AO
MK01	36	16,67%	13,89%	42	41	42	42	42	41,9	0,00	0,30
MK02	24	33,33%	20,83%	32	29	35	35	33,2	31,5	0,98	1,69
MK03	204	0,00%	0,00%	204	204	204	204	204	204	0,00	0,00
MK04	48	37,50%	29,17%	66	62	68	67	67	65,4	0,45	1,56
MK05	168	5,36%	6,55%	177	179	187	184	182	180,3	2,79	1,35
MK06	33	124,24%	109,09%	74	69	82	75	77,3	71,7	2,72	1,68
MK07	133	18,05%	21,05%	157	161	171	173	165,9	164,1	4,21	3,48
MK08	523	0,00%	0,00%	523	523	523	523	523	523	0,00	0,00
MK09	299	7,02%	9,03%	320	326	339	340	332,2	331,9	6,00	4,53
MK10	165	49,09%	56,97%	246	259	264	283	253,3	271,7	5,75	8,17
Průměrná vzdálenost od optima		29,13%	26,66%					Průměrná S.odchylka		2,29	2,27

Tab. 6-49: Porovnání EA a AO modelů FJSSP – rozvrhy bez zpoždění

Model	Optimum	Odchylka od optima EA	Odchylka od optima AO	Min EA	Min AO	Max EA	Max AO	Průměr EA	Průměr AO	S.odchylka EA	S.odchylka AO
MK01	36	16,67%	16,67%	42	42	42	42	42	42	0,00	0,00
MK02	24	29,17%	33,33%	31	32	34	35	32,3	33,2	0,78	0,98
MK03	204	0,00%	0,00%	204	204	204	204	204	204	0,00	0,00
MK04	48	37,50%	37,50%	66	66	67	68	66,8	67	0,40	0,45
MK05	168	4,76%	5,36%	176	177	179	187	177,8	182	0,98	2,79
MK06	33	139,39%	124,24%	79	74	85	82	81,8	77,3	1,78	2,72
MK07	133	21,80%	18,05%	162	157	170	171	167,3	165,9	2,87	4,21
MK08	523	0,00%	0,00%	523	523	523	523	523	523	0,00	0,00
MK09	299	5,02%	7,02%	314	320	322	339	317,8	332,2	3,03	6,00
MK10	165	55,15%	49,09%	256	246	264	264	259,6	253,3	2,15	5,75
Průměrná vzdálenost od optima		30,95%	29,13%					Průměrná S.odchylka		1,20	2,29

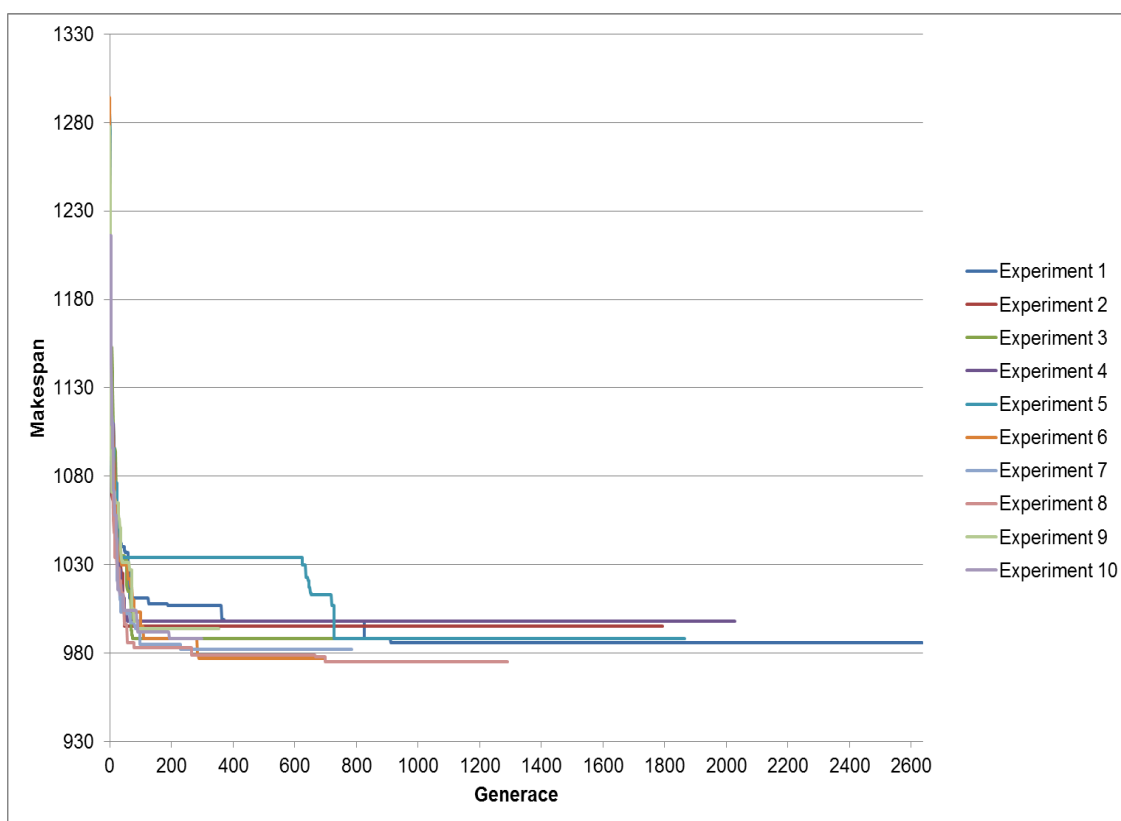
V případě modelů FJSSP nachází AO srovnatelné či jen o málo lepší výsledky než u EA. Na rozdíl od modelů JSSP mají mírně lepší hodnoty účelové funkce aktivní rozvrhy než rozvrhy bez zpoždění.

Malé odchylky AO od EA přes větší časovou dotaci u většiny EA modelů FJSSP a JSSP jsou způsobeny adaptabilním mechanismem stanovování počtu generací a velikosti optimalizace. V AO se snoubí prohledávání do šířky (v případě velkých populací se prohledává velká množina řešení) a do hloubky (rychlá konvergence k optimálnímu řešení).

Mechanismus na počátku, kdy je nastavena malá populace (10 jedinců), rychle konverguje k lokálnímu extrému. Během optimalizace se však populace neustále zvětšuje v závislosti na kvalitě nově nacházených jedinců. Populace díky tomu nemá možnost úplně konvergovat k jednomu řešení (lokální extrém).

Pokud se nedaří získaný lokální extrém prolomit, snižuje se velikost populace, čímž se zvětšuje pravděpodobnost výměny genu těch nejlepších jedinců.

Výsledkem jsou relativně malé rozdíly mezi výsledky získanými pomocí EA a AO a to v kratším čase, než je tomu u EA (graf 6- 17).



Graf 6-17: 10 experimentů modelu FT10 pomocí AO

6.5 Dílčí závěr č.4 – Zhodnocení výsledků heuristických algoritmů aplikovaných na etalonových modelech

Pro hodnocení výkonnosti jednotlivých heuristických algoritmů bylo použito celkem 20 etalonových modelů rozvrhovacích problémů JSSP a FJSSP. Na každý z těchto modelů byly postupně aplikovány metody konstruktivního algoritmu, lokálního prohledávání, evolučního algoritmu a kaskádové automatické optimalizace, která používá předchozí zmiňované metody.

Konstruktivní algoritmus (CA), jako nejpoužívanější metoda, byl hodnocen na základě jeho schopnosti dosáhnout optima (účelová funkce makespan) a času potřebného pro generování výsledného rozvrhu.

Výsledky testování potvrdily, že ačkoli je CA velmi rychlý algoritmus, nedokáže pomocí aplikovaných prioritních pravidel (SPT, LPT, MWKR, FIFO a RS) nalézt optimální výsledky testovaných etalonových modelů.

Oproti tomu analýza lokálního prohledávání (LS) ukázala, že navržený mechanismus LS při delším výpočetním čase dokáže v některých případech nalézt optimum či řešení blízka optimu. Ve většině případů však LS nedokázalo zlepšit počáteční řešení dané CA. Po převedení obecných časových simulačních jednotek modelu na minuty a následném porovnání dosažených úspor oproti CA lze konstatovat, LS lze s výhodou použít v případě, že je dostupný čas na optimalizaci před realizací výrobního plánu.

Hlavní důraz v případě analýzy evolučního algoritmu byl pak kladen na nalezení vhodného nastavení optimalizačních parametrů selekce, křížení a následné hybridizace.

Výsledky testů na vybraných obtížných modelech FT10 a MK10 ukázaly, že v některých případech má nastavení parametru křížení vliv na výslednou hodnotu účelové funkce. Při nastavení pravděpodobnosti křížení na 10% poskytoval EA výsledky nejbližší optimu, avšak v případě modelu FT10 se hodnoty účelové funkce při jiných nastaveních statisticky příliš nelišily.

Oproti tomu vliv selekce byl patrný na obou modelech. Tak, jak se předpokládalo, elitářská selekce bez nápravných mechanismů zabránění konvergence celé populace k jednomu řešení poskytovala hodnoty nejvzdálenější optimu. Z dvojice selekcí PvCH, používající turnajový princip a ECC, založené na kontrole klonů, byla pro další aplikaci EA vybrána ECC na základě lepších výsledků makespan a silné pozitivní (ve smyslu hodnot bližších optimu) interakce s 10% křížením.

Analýzy výsledků testování koeficientu hybridizace poukázaly na možný vliv na hodnotu účelové funkce makespan. Použitím hybridizace lze dosáhnout lepších výsledků než bez ní a to jak s ohledem na nalezené nejlepší řešení, tak na rozdíly mezi jednotlivými běhy experimentu. Na základě časové náročnosti optimalizace a dosažených hodnot funkce makespan je doporučeno pro navržený EA zvolit koeficient hybridizace 0,1 (na 10% populace je použita hybridizace).

Posouzení výkonnosti (dosažení výsledků měřené makespanem v čase) navrženého nastavení optimalizačních parametrů je jako v případě LS hodnoceno kvalitou výsledků neboli schopností dosáhnout optima. EA je schopen nalézt kvalitnější výsledky než LS. V případě JSSP modelů bylo dosaženo optima ve dvou případech,

ostatní výsledky byly v průměru 5% od optimálního výsledku, což lze pokládat za výsledky velmi blízké optimu. Optimalizace modelů FJSSP byla, jak se předpokládalo, vzhledem ke složitosti problému méně úspěšná. Ačkoli bylo ve 2 případech dosaženo optimální hodnoty funkce makespan, v průměru se výsledky lišily od optimální hodnoty o cca 30%. Přesto lze hodnotit optimalizaci pomocí EA jako přínosné. V porovnání s konstruktivními algoritmy jsou nacházena v průměru o 10%-20% lepší řešení.

Z porovnání celkového přínosu optimalizace vyplývá podobně jako u LS, že při použití EA lze předpokládat dosažení značné úspory času oproti klasickým CA metodám. Je však nutné brát na zřetel časový náklad na optimalizaci. Přes to, že v přibližně polovině případů EA přineslo úsporu (makespan převedený na minuty) okolo 1-2 hodiny, v ostatních případech se investice do optimalizace nevyplatila. Příkladem může být model SW11, kde došlo cca k 6 hodinové úspoře času, avšak optimalizace trvala více než 8 hodin. V takové situaci lze doporučit aplikaci EA pouze, pokud dochází k optimalizaci výrobního rozvrhu před zavedením plánu do výroby. V takovém případě má optimalizace na modelech, které jsou podobné prezentovaným etalonovým modelům JSSP a FJSSP, značný přínos.

Kompromisem mezi metodami CA, LS a EA je pak jejich postupné použití v rámci navržené automatické optimalizace, která na základě času dostupného na optimalizaci určuje použití jednotlivých heuristických metod. Výhodou této metody je využití poznatků z testování EA, díky kterému jsou nastaveny parametry automatické optimalizace s výjimkou velikosti populace a počtu generací. Tyto parametry, které mají zásadní vliv na kvalitu výsledku a délku optimalizace, jsou určovány pomocí modifikovaného adaptivního mechanismu. Testy se stanovenou délkou optimalizace na 15 minut ukázaly, že tato metoda může nalézt přibližně stejně kvalitní výsledky v porovnání s EA.

Z provedených analýz se jeví jako vhodné použití principu automatické optimalizace pro dílenské řízení výroby. Hlavním přínosem je pak možná aplikace metody v nevýrobním – plánovacím období, kde lze předpokládat vzhledem k většímu dostupnému času pro plánování větších zlepšení účelových funkcí (počet zpožděných zakázek či makespan).

Posledním krokem ke splnění hlavního cíle této práce je tedy ověření přínosů testovaných algoritmů na modelech reálné výroby pomocí navržené metodiky.

7 OVĚŘENÍ METODIKY VYUŽITÍ ROZVRHOVACÍCH METOD NA MODELECH REÁLNÉ VÝROBY

Tato kapitola má za úkol ověřit metodiku a navržené algoritmy z přecházejících kapitol a uvést příklady možného využití pokročilých rozvrhovacích metod.

Pro testování byly zvoleny společnosti z oblasti strojírenské výroby TOS VARNSDORF a.s., který se zabývá především výrobou horizontálních frézovacích a vyvrtávacích strojů a obráběcích center a MZ Liberec a.s., který se zabývá zejména zařízeními pro rozvod technických plynů a plynů používaných v nemocnicích a podobných zařízeních.

Výběr těchto firem také ovlivnila rozdílná zkušenost se systémy plánování a řízení výroby, kdy TOS VARSDORF a.s. má bohaté zkušenosti se systémy APS a MZ Liberec a.s. o zavedení těchto systémů teprve uvažuje.

7.1 TOS VARNSDORF a.s.

Společnost TOS VARNSDORF se již od roku 1903 zabývá strojní výrobou. Hlavními artikly společnosti jsou:

- Stolové a deskové horizontální vyvrtávačky.
- Horizontální obráběcí centra včetně vysokorychlostních (HSM).
- Příslušenství ke strojům (obráběcí hlavy, stoly, upínací zařízení atd.).

Společnost má v současné době cca 500 zaměstnanců (konec roku 2012) a její roční obrat činí cca 2,5 miliardy Kč. Dle těchto měřítek lze považovat podnik za velký.

Přestože je tato práce zaměřena na řízení výroby SME podniků, byla firma vybrána na základě požadavku na optimalizaci dílenského plánování (viz následující kapitoly) jednoho ze tří hlavních segmentů (lehká výroba, těžká výroba a montáž) a to lehké mechaniky a výroby obráběcích hlav označovaných jako lehká výroba či kovodílna.

7.1.1 Definování požadavku na plán – rozvrh

Hlavní požadavek na dílenský plán – rozvrh vychází z obtížného plánování části výroby, která má za úkol dodávat vřeteníky, obráběcí hlavy a lehké díly na finální montáž obráběcích strojů. Hlavním problémem v této oblasti je kumulace fronty práce, která je definovaná lhůtovým plánem v návaznosti na ostatní prvky vstupující na finální montáž. Krátkodobým plánovacím horizontem je jeden týden, s výhledem na 3 měsíce (dlouhodobý plán). Průběžná doba výroby v závislosti na naplnění kapacit a složitosti výrobků činí cca. 3 měsíce. Krátkodobý plán je definován frekvencí přepočtu tohoto plánu (obvykle jednou týdně), který musí reflektovat termíny odvádění výroby na montáž.

Mezi hlavní omezení výroby v lehké kovodílně patří:

- Termín uvolnění materiálu do výroby – je dán plánem nákupu a lhůtovým plánem.
- Strojní zařízení – jedná se o stroje na lehké kovodílně a o stroje mimo tuto dílnu. Stroje mimo dílnu jsou plánované v rámci vnitřní kooperace. Jedná se zvláště o

obtížně plánované technologie povrchové a tepelné úpravy jako je kalení, žihání a černění.

- Směnný systém – skládá se ze dvou rozdílných směnných systémů (8 hodinového a 12 hodinového cyklu), které přímo závisí na délce operací. Hlavním problémem je synchronizace těchto směnných systémů s ohledem na montáž.
- Doba zpracování výrobních dávek a trajektorie dílů výrobním systémem – je dáno technologickým postupem.
- Manipulace a transport – manipulací se rozumí pohyb dílců mezi stroji uvnitř střediska, transportem pak přesun dílců v dávkách a na větší vzdálenosti (např. vnitřní kolejovou dopravou).
- Poruchy zmetky a údržba strojů – toto omezení je vázáno na seřizovací a kusové časy v poměru daném historickou zkušeností a kvalifikovaným odhadem.

Dalším krokem metodiky je pak kvantifikování navržené účelové funkce a omezení modelu.

7.1.2 Stanovení měřitelných ukazatelů

Ve fázi stanovení měřitelných ukazatelů se kvantifikují definované problémy a omezení tak, aby bylo možné analyzovat reálná zlepšení.

Vzhledem k problému vzrůstající fronty práce je pro optimalizaci dílenského rozvrhu zvolena účelová funkce **makespan**. Ta odráží celkový čas dohotovení všech zakázek daných lhůtovým plánem, který vychází z globálního dlouhodobého plánu. Jako kontrolní mechanismus pak slouží **opoždění jednotlivých zakázek** na finální montáž.

Nadefinovaná omezení se kvantifikují následujícím způsobem:

- Termín uvolnění materiálu do výroby – termíny jsou stanoveny obvykle s přesností na dny. Vzhledem k přesnosti algoritmu se termín uvolnění materiálu do výroby určuje s přesností na minuty a to začátkem první směny dne dle plánu navážky do výroby. Velikost výrobní dávky je i přes nákup v hmotnostních jednotkách převedena pro jednoduchost na kusy.
- Strojní zařízení – rozvrhují se obvykle s přesností na hodiny. Pro zachování jednotného formátu vzhledem k automatizovaným pracovištím je nejmenší časovou jednotkou pro rozvrhování vteřina. Časový fond stroje je pak závislý na typu technologie. V případě klasické kusové či dávkové výroby odráží časový fond zvolený typ směnnosti. Pokud se využívají ve vnitřní kooperaci skupinové technologie (např. černění, žihání, kalení) a tudíž není možné rozvrhovat výrobní dávky vzhledem k jejich rozpadu dle vlastností technologie, počítá se s tímto omezením jako s transportem. Celková doba transportu pak odpovídá lhůtovému plánu resp. průběžné době výroby.
- Směnný systém – pro potřeby dalšího použití v jiných modelech strojírenské výroby je třeba, aby algoritmus umožňoval (více viz analýza současného stavu-seřizování) následující směnné systémy:
 - Dvousměnný 12 hodinový s jednou či dvěma směny a s volitelným začátkem první směny.

- Třísměnný 8 hodinový systém s jednou, dvěma či třemi směnami a s volitelným začátkem první směny.
- Čas zpracování výrobních dávek a trajektorie dílů výrobním systémem - časy zpracování výrobních dávek jsou závislé na době seřízení (v sekundách) a době výroby jednoho kusu (sekundy). Trajektorie je určována (stejně jako časy výrobních a nevýrobních operací) technologickým postupem s rozlišováním jednotlivých strojů, a ne jenom výrobních středisek. Hlavním důvodem je co největší rozlišení dílenského plánu, aby bylo možné vytvořit plán i pro záměnná pracoviště.
- Manipulace a transport – délka transportu je definovaná časem (ne vzdáleností) potřebným pro převezení materiálu mezi stroji či vstupním a výstupním skladem.
- Poruchy, zmetky a údržba strojů – není uvažováno jako primární omezení.

7.1.3 Analýza současného stavu

Vzhledem k definovaným cílům a stanoveným měřitelným ukazatelům je provedena analýza současného stavu v těchto krocích:

- 1) Existující plánovací metody - Společnost TOS VARNSDORF a.s. v současné době používá jako hlavní plánovací systém I2 Factory Planner. Tento APS systém nahradil dřívější systém plánující pouze pomocí metod založených na MRP II, tzn. na základě kapacitní a lhůtové bilance. Výhodou I2 Factory Planner je pokročilé plánovací jádro inspirované systémem OPT, tedy řízením pomocí úzkých míst. Na základě poptávky a současných skladovacích zásob metodou Capable-to-Promise rozvrhuje výrobní operace.

Důležitým rozhodnutím při použití metody řízení úzkých míst je nalezení úzkého místa. V případě této společnosti je definované jako úzké místo jedno z pracovišť na těžké kovovýrobě, což vyhovuje ve většině situací vznikajících nad plánem. Pro toto úzké místo se vytváří rozvrh, který je optimalizován na základě zákaznických termínů a nákladů na seřizování. Hlavní plánovací algoritmus mění sekvenci výrobních dávek tak, aby co nejvíce minimalizoval čas investovaný do seřizování a zároveň dodržel termíny dané zákazníkem. Na základě tohoto rozvrhu se pak řídí pracoviště předcházející v technologickém postupu pomocí tahu, následná pracoviště či pracoviště nekritická pak odvozují svůj plán na základě lhůt.

Z analýzy materiálového toku celým systémem vyplynulo, že v závislosti na výrobním programu (požadavku zákazníka) se definované úzké místo přesouvá mimo těžkou kovovýrobu. Vzhledem ke kumulaci práce na středisku lehké kovovýroby je předpoklad, že se úzké místo přesouvá právě na toto středisko.

Toto středisko je v současné době plánováno na základě rozvrhu pracovišť těžké techniky, tzn. fronta práce je tvořena před pracovišti pomocí odhadu průběžné doby výroby. Ta je kalkulována jako suma požadavků na strojní zařízení dle technologického postupu.

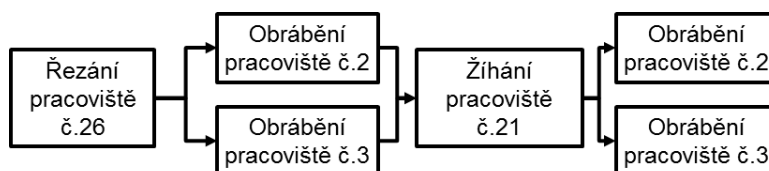
- 2) Časová náročnost plánování a rozvrhování - Rozdělení práce na záměnné technologie a sekvenci výrobních dávek během dne určuje dílenský plánovač (mistr) na základě zkušenosti (předchozí výroby na strojích, termínů

odvádění na montáž atd.). Dílenský mistr před začátkem směny investuje cca 15-45 minut (v závislosti na složitosti situace na dílně) do rozvržení výroby. V průběhu směny pak provádí korekce na základě nenadálých situací.

- 3) Způsob odvádění hotové výroby – hotová výroba se odvádí po dokončení poslední operace výrobní dávky pomocí čárového kódu na výrobním příkaze vázaného na související výrobní zdroj.
- 4) Výběr hlavních představitelů – Hlavním sortimentem tohoto střediska jsou obráběcí hlavy a lehká technika vstupující na montáž. S ohledem na složitost výrobků jsou vybrány ve spolupráci s pracovníky tohoto střediska tři typy hlav, které se vyrábí nejčastěji a zároveň se liší složitostí jejich výroby:
 - HUR – jedná se o výrobně nejjednodušší hlavici, jejíž sestava obsahuje 14 hlavních komponentů. Tento výrobek je zpracováván postupně na 5-11 strojích.
 - HUI – hlavice obsahující 19 hlavních komponent, které se postupně zpracovávají na 6-24 strojích.
 - HOI - nejsložitější hlavice obsahující 51 hlavních komponent, které se zpracovávají postupně na 3-25 strojích.

Model tedy obsahuje celkem 84 výrobních komponentů.

- 5) Druh výroby - Na základě analýzy technologických postupů vybraných hlav je pro rozvrhování určen model FJSSP jako reprezentativní. To zvláště s ohledem na zaměnitelnost pracovišť a trajektorii výrobním systémem, kde se často vyskytují prvky JSSP problému. Zakázky se v některých případech navrací na pracoviště, na kterém již byly zpracovávány. Typickým příkladem jsou zámečnické operace, pracoviště kontroly či obráběcí stroje, na které se výrobky vrací po tepelném zpracování (viz obr. 7-1).



Obr. 7-1: Příklad trajektorie výrobním systémem TOS VARSDORF a.s.

6) Omezení výroby

- Strojní zařízení - model výrobního systému vycházející z technologického postupu, který zpracovává uvedené hlavice, obsahuje:
 - 19 obráběcích strojů, z nichž dva páry strojů jsou záměnné. Jedná se tedy o flexibilní typ modelu.
 - Jedno pracoviště, na kterém se provádí zámečnické práce.
 - 15 pracovišť v interní kooperaci jako jsou – žíhání, stolní tryskání, natírání, kalení, černění, 3D kontrola atd.
- Směnnost - Pracoviště pracují v 8 hodinovém směnném systému s 1-3 směny, kde první směna začíná v 6:00.
- Transport – délka transportu je určena vzdáleností a způsobem dopravy mezi pracovišti. Pomocí transportu se také zohledňuje výroba na skupinových technologiích v rámci interní kooperace.

7) Množství zakázek, operací a čas jejich zpracování –

- Množství zakázek v modelu odráží poptávku k I.Q. 2012:
 - HUR 6 ks
 - HUI 6 ks
 - HOI 1 ks
- Délka trvání výroby jednoho komponentu na jednom pracovišti může dosáhnout až cca 1900 minut (kusový čas).
- Doba seřízení stroje na výrobní dávku může trvat až cca 700 minut. Velikost výrobních dávek a celkový objem výroby odráží poptávku z období prvního kvartálu roku 2012.

8) Konzistence dat –

- Identifikace zakázky – výrobní příkazy mají jedinečnou identifikaci pro každou z výrobních zakázek dle plánu finální výroby. Pro potřeby rozvrhování v navržené aplikaci dostává každý díl sestavy jednoznačnou identifikaci danou pořadovým číslem.
- Termíny uvolnění zakázek – dle lhůtového plánu s přesností na hodiny v případě nekritických pracovišť. Kritické pracoviště pak dle detailního rozvrhu. Předpokládá se, že po aplikaci rozvrhování dojde ke změně termínů uvolnění zakázek vzhledem k rozvrhovacím konfliktům a vytížením pracovišť. To je způsobeno lhůtovým rozvrhem, který reflektuje průběžnou dobu výroby, ale ne více výrobních požadavků na výrobu na jedno pracoviště v jeden čas).
- Termíny odvedení zakázek – na střediscích lehké kovovýroby je uváděn termín odvádění na montáž s přesností na hodiny. Tak jako u termínu uvolnění zakázky dojde ke korekci těchto termínů.
- Množství (požadavek a výrobní dávka) – odráží požadavek zákazníka. Na sklad se plánují pouze nakupované položky vstupující na montáž dle plánu nákupu.

- Identifikace zdrojů – každý zdroj má jedinečnou identifikaci včetně záměnných strojů (43-41260-0 = číslo střediska - číslo stroje - pořadové číslo stroje stejné technologie). Pro potřeby rozvrhovacího modelu je každému stroji přiřazeno pořadové číslo.
- Časy kusové, seřizovací a transportní – v případě transportních časů se vychází ze vzdálenosti a expertního odhadu délky trvání dle manipulační techniky. Časy kusové a seřizovací vychází z technologického postupu z informačního systému. Pro potřeby rozvrhování je však nutné provést revizi některých operací. Příkladem může být operace kontroly, u které se často vyskytuje nulový čas seřízení i nulový kusový čas. To vychází z financování operace kontroly z jiného nákladového střediska. Pro tyto druhy operací jsou pro potřeby rozvrhování tyto informace doplněny do modelu. Dalším příkladem jsou operace tepelné a povrchové úpravy, u kterých nejsou uvedeny seřizovací a kusové časy. Důvodem je rozpad výrobních dávek dle možností technologie. Proto je u těchto operací zvolen princip lhůtového rozvrhování pomocí transportních operací. Délka transportu pak vychází z průběžné doby výroby.

Na základě analýzy výrobního systému vznikla nutnost doplnit do klasického rozvrhovacího algoritmu tato omezení:

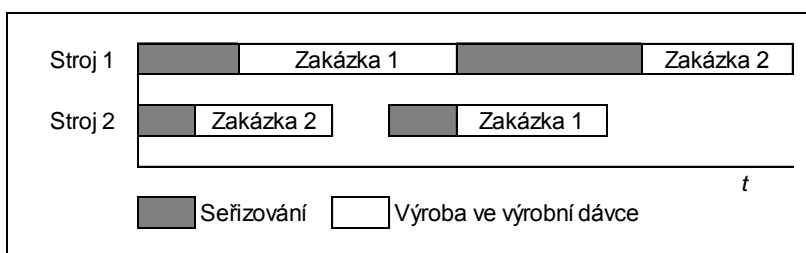
- Seřizován,
- Směnnost
- Transport

Seřizování

Modifikovaný algoritmus Gifflera a Thomsona pro aktivní rozvrhy a rozvrhy bez zpoždění umožňuje zpracovávání operací jednu po druhé, kde výběr následující operace závisí na nejdříve možném začátku či konci operace. Tyto termíny jsou zásadně ovlivněny velikostí výrobní dávky a časem seřízení [171].

V případě použití jednoduchého seřizovacího času, kde je potřebný alespoň jeden díl z výrobní dávky pro provedení seřízení, je začátek výrobní dávky dán prvním dostupným časem zpracování dávky (nejpozdější čas dostupnosti dávky a zároveň stroje), ke kterému se přičte suma kusových časů a času seřízení.

Složitější situace nastává, pokud lze seřídít na výrobní dávku bez existence výrobního kusu pomocí přípravků, nástrojů standardizace výrobků atd. (obr. 7-2 zobrazuje problém dvou zakázek zpracovávaných na dvou strojích s opačnou sekvencí zpracovávání na zdrojích). V takovém případě je nutné modifikovat startovní časy výrobních zakázek, kde jsou omezení dostupnosti zakázky a stroje brána samostatně. Seřizování lze začít v případě, že je dostupný stroj. Začátek zpracování zakázky je pak omezeno jak dostupností stroje, tak dostupností předcházejících operací.



Obr. 7-2: Ganttův graf seřizování bez potřeby kusů z předchozí operace

Startovní a koncové časy výrobních dávek jsou dále ovlivněny směnností.

Směnnost

Reflektování omezení v rozložení směn vychází z praktických omezení ve výrobě, kde hlavními údaji o směnnosti jsou:

- směnnost (např. dvou či tří směnný systém),
- počet směn ve dni,
- délka směny,
- délka přestávky,
- začátek první směny.

Algoritmus úpravy začátku a konce omezení dle směnnosti obsahuje hlavní tři části.

První část stanovuje možný relativní nejdříve možný začátek operace s ohledem na dostupnost zakázky pro každý stroj zvlášť. Tento čas musí respektovat nastavení směnnosti, to znamená, že operace nemůže začít v přestávce či mimo pracovní směnu. Algoritmus tedy posunuje možný začátek operace na nejdříve možný čas posunem tohoto termínu do budoucnosti (takzvaný posun vpravo). Absolutní čas se pak současně upravuje se změnou času relativního.

V druhé části se stanovuje začátek operace na základě nejdříve možného začátku operace, stroje a druhu seřízení. Algoritmus v této části posouvá startovací čas do minulosti (resp. vlevo) na základě délky trvání a druhu seřízení. Počínaje relativním časem začátku operace algoritmus iterativně na základě přestávek a směnnosti odečítá seřizovací čas, dokud nějaký zbývá, či nedosáhne termínu, kdy je naposledy dostupný stroj.

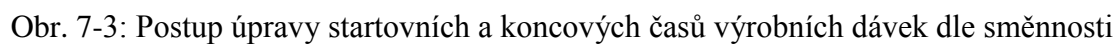
V třetí fázi na základě stanoveného začátku výrobní dávky (či operace) se iterativně (podobně jako v diskretní simulaci) spotřebovává čas na provedení seřízení a zpracování výrobní dávky.

Jedná se tedy zjednodušeně o formu zpětně-dopředného plánování s ohledem na omezení směnnosti a přestávek (obr. 7-3).

Algoritmus pracuje s následujícími proměnnými:

- e_t – absolutní počáteční začátek operace - zakázky (např. 22:30 1.1.2009)
- e_{tr} – absolutní počáteční začátek operace - zakázky ve vybraných časových jednotkách (e.g. 30 [min])
- f_t – absolutní čas konce zakázky (operace)
- f_{tr} – relativní čas konce zakázky (operace)
- o_k – vybraná operace (zakázka)
- p_k – délka zpracovávání o_k
- se_k – délka seřízení o_k
- m_k – stroj zpracovávající o_k
- m_t – absolutní čas nejdříve možného začátku operace (zakázky) o_k
- m_{tr} – relativní čas nejdříve možného začátku operace (zakázky) o_k
- sf_t – délka směny
- br_t – délka přestávky

$\text{sft}_m\{\}$ – dostupná směna na stroji



Transport

Manipulace a transport je jedním z hlavních omezení vyskytujícím se v každém typu výroby. Zahrnutí tohoto omezení bývá často problematické vzhledem k tomu, že až na výjimky nebývá proces manipulace zanalyzovaný do té míry, jako tomu bývá u procesních či seřizovacích časů. V případě TOS VARNSDORF a.s. jsou manipulační časy odvozené ze vzdáleností jednotlivých středisek (strojů) a kvalifikovaného odhadu vycházejícího ze zkušenosti. Samotná algoritmizace manipulace v modelu není složitá. Nejkratší možný začátek zpracování zakázky je odvozen z konce předcházejících operací, ke kterému se připočítá manipulační čas vycházející z transportní matice (tab. 7-1).

Tab. 7-1: Ilustrace transportní matice

Střediska	Vstupní sklad	1	2	3	4	5	6	Expediční sklad
Vstupní sklad	480	480	480	480	480	480	480	480
1	480	20	25	24	48	68	78	480
2	480	25	20	80	32	64	70	480
3	480	24	80	20	50	63	90	480
4	480	48	32	50	20	78	120	480
5	480	68	64	63	78	20	80	480
6	480	78	70	90	120	80	20	480
Expediční sklad	480	480	480	480	480	480	480	20

Takto aplikovanou manipulaci lze s výhodou použít i v případech, kde není možné použít rozvrhovací model pro jednotlivé výrobní dávky. Jedná se např. o operace tepelné (žihací pece) či povrchové úpravy (černění), kde se dávky rozpadají na vnitřní dávky střediska vzhledem k omezením vycházejícím z podstaty technologie (např. žihání ocele a litiny - prostorové a teplotní omezení pece).

Další modifikace modelu dle požadovaných omezení probíhá v rámci kroku testování modelu a navržených optimalizačních metod.

7.1.4 Testování modelu pomocí navržených rozvrhovacích metod

Prvním krokem je testování základního modelu. To zahrnuje kontrolu logiky omezení (zejména seřizování, směnnosti a transportu) pomocí konstruktivního algoritmu. Pro toto testování je zvolen nejjednodušší výrobek obráběcí hlava HUR s 14 vstupujícími komponenty, na kterých je provedeno celkem 161 výrobních operací.

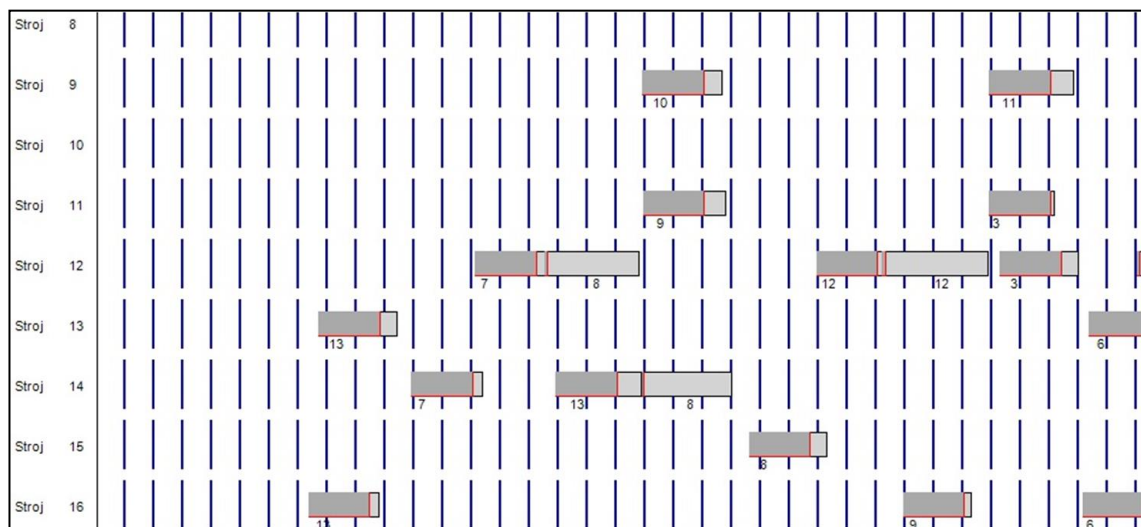
V prvním kroku je provedena ruční kontrola začátku a konce operací vybraného rozvrhu (viz tab. 7-2).

Tab. 7-2: Náhled kontroly 1/3 modelu 1. komponentu hlavy HUR

Číslo operace	Číslo stroje	Start operace relativně [min]	Konec operace realativně [min]	Sekvence celková	Procesní čas [min]	Start operace absolutně [min]	Konec operace absolutně [min]	Seřizování [min]	Transport [min]
0	26	719	726	0	6	2.12.2011 17:59	2.12.2011 18:06	1	720
1	4	9060	10182	39	102	8.12.2011 13:00	9.12.2011 7:42	1020	8640
2	25	13062	13068	57	6	11.12.2011 7:42	11.12.2011 7:48	0	2880
3	4	13380	14502	66	102	11.12.2011 13:00	12.12.2011 7:42	1020	720
4	25	17382	17562	91	180	14.12.2011 7:42	14.12.2011 10:42	0	2880
5	16	17700	18948	97	228	14.12.2011 13:00	15.12.2011 9:48	1020	720
6	36	19668	19728	102	60	15.12.2011 21:48	15.12.2011 22:48	0	720

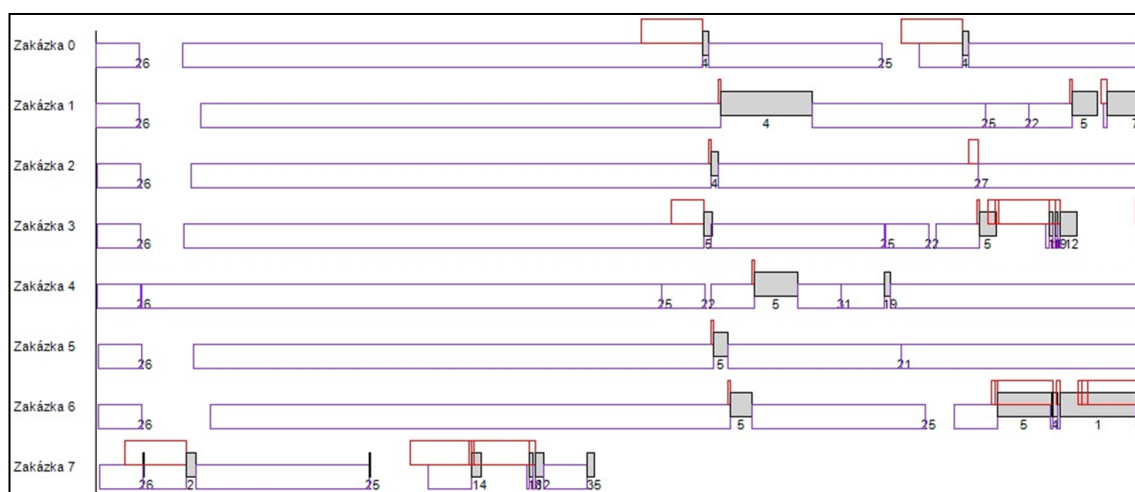
Po provedení ruční kontroly operací je provedena namátková vizuální kontrola Ganttova diagramu z pohledu přes zakázky (na ose y jsou zakázky) a přes stroje (na ose x jsou stroje). Z pohledu přes stroje (obr. 7-4) lze ověřit, zda se nepřekrývá seřizování

(šedé bloky s červeným podtržením) a operace (šedivé bloky) na jednotlivých strojích. Dále tato kontrola slouží k ověření směnnosti (směny jsou odděleny modře přestávkami)



Obr. 7-4: Ganttův diagram - ukázka pohledu přes stroje

Z pohledu přes zakázky (obr 7-5) pak probíhá kontrola seřizování (červeně) transportu (modře) a průběh výrobních dávek (šedivě).



Obr. 7-5: Ganttův diagram - ukázka pohledu přes zakázky

Po vizuální kontrole je provedena analýza časové náročnosti a odhad kombinatorické složitosti modelu.

Výpočet účelové funkce makespan pomocí konstruktivního algoritmu (CA) spolu s prioritními pravidly SPT, LPT, MWKR, FIFO a RS (deset rozvrhů) s použitím aktivních rozvrhů a rozvrhů bez zpoždění trvá 4,9 s.

Vzhledem k relativně malé zátěži výrobního systému (modelu) zakázkami je kombinatorická složitost malá (náhodně generované rozvrhy mají stejnou hodnotu). Přesto lze vidět určitý potenciál pro zlepšení, kdy už na úrovni konstruktivního algoritmu jsou patrné rozdíly – jsou nalezena různá řešení (tab. 7-3) v rozsahu jednoho dne (1440 minut).

Tab. 7-3: Hodnoty účelové funkce makespan generované pomocí CA 1/3 modelu TOS VARNSDORF a.s.

Prioritní pravidla	Aktivní rozvrhy [min]	Rozvrhy bez zpoždění [min]
SPT	63036	63036
LPT	63036	64476
MWKR	63036	64476
FIFO	63036	63036
RS	63036	64476

V druhém kroku testování modelu TOS VARNSDORF a.s. je přidána zakázka na obráběcí hlavy HUI. V první fázi je testován model bez vazby na termíny uvolňování materiálu - odhad kombinatorické složitosti. V této fázi je již model optimalizován pomocí kaskádové automatické optimalizace s časem pro optimalizaci 15 minut. Tento čas vychází jak z požadavků pracovníků společnosti, tak z odhadu času (na základě předchozích testů) potřebného pro otestování metod CA LS i EA.

Model obsahující obráběcí hlavy HUR a HUI, jejichž celkový počet výrobních dávek nutných ke zpracování je 408, kaskádový algoritmus testoval metodou CA a EA, přičemž metodu LS vynechal vzhledem k předpokládané vysoké časové náročnosti (odhad na základě trvání CA).

Jak je patrné z tabulky 7-4, kombinatorická složitost modelu se zvýšila. To vyplývá z různých hodnot účelové funkce makespan, kde plány generované pomocí pravidla RS (a nejen pomocí něho) nabývají různých hodnot (RS_b – nejlepší hodnota makespan, RS_w nejhoší a RS_a průměrná).

Tab. 7-4: Hodnoty makespan dle automatické optimalizace 2/3 modelu TOS VARNSDORF a.s.

Metoda	Prioritní pravidla	Aktivní rozvrhy [min]	Rozvrhy bez zpoždění [min]
CA	SPT	96246	71676
	LPT	72991	76056
	MWKR	80406	70236
	FIFO	71766	76056
	RS_b	73176	73122
	RS_w	80406	82402
	RS_a	76227	75258
EA		65922	

Z porovnání výsledků mezi jednotlivými prioritními pravidly a výsledku EA vyplývá, že pokud se vhodně zvolí náběh materiálu a termín odvedení zákazníkovi, může hodnota makespan razantně klesnout. Rozdíl mezi nejhorším výsledkem CA (SPT aktivní rozvrhy) a výsledkem EA činí více než 30 000 minut, což lze převést při 1440 minutách v jednom dni na úsporu 21 pracovních dnů. Při porovnání s nejlepším výsledkem CA (MKWR – rozvrhy bez zpoždění) úspora činí necelé 3 dny. Pokud se přihlédne k nejčastěji používanému prioritnímu pravidlu (FIFO v kombinaci s rozvrhy bez zpoždění), úspora činí cca 7 pracovních dní, resp. cca 13% úsporu času.

V následujícím kroku jsou do modelu zavedeny termíny uvolňování materiálu do výroby pro porovnání reálnosti generovaných rozvrhů a lze tedy zkontrolovat reálnost nastavení termínů odvádění zákazníkovi.

Pro porovnání zákaznických termínů s termíny odvedení hotové výroby je použito pravidlo FIFO s rozvrhy bez zpoždění, které se používá ve společnosti při operativním řízení výroby.

Z testu modelu s vlivem uvolnění materiálu do výroby vyplývá nevhodné nastavení termínů (viz příloha č. 1), kde hodnota makespan ve všech případech (CA i EA) dosahuje stejné hodnoty. Toto se potvrzuje porovnáním požadovaných zákaznických termínů s reálnými termíny, kdy v některých případech je možné zakázky odvést o 30 pracovních dnů dříve a jiné zakázky jsou odvedené o více než 10 pracovních dnů později.

Vzhledem k použití účelové funkce makespan je pravděpodobné, že zákaznické termíny se budou od rozvržených termínů lišit. Následující test po přidání posledního z výrobků proto obsahuje optimalizaci podle hodnoty makespan a zákaznických termínů, na základě kterého jsou po konzultaci s dílenskými plánovači lehké kovovýroby a finální montáže stanoveny nové termíny uvolnění materiálu do výroby.

Vzhledem k původnímu nerealistickému nastavení lhůtového plánu daného stávajícím informačním systémem je stanoven termín náběhu výroby a termín odvedení zákazníkovi pro všechny obráběcí hlavy stejně (náběh 5.12.2011 a zákaznický termín 5.1.2012). Tento termín je stanoven na základě průběžné doby výroby pro všechny 3 hlavy (2-3 měsíce).

Následuje test plného modelu. Tak jako v předchozím testování je použita metoda automatické optimalizace s účelovou funkcí makespan (tab. 7-5) a čas dostupný pro optimalizaci je 15 minut.

Tab. 7-5: Hodnoty makespan dle automatické optimalizace plného modelu TOS VARNSDORF a.s.

Metoda	Prioritní pravidla	Aktivní rozvrhy [min]	Rozvrhy bez zpoždění [min]
CA	SPT	109206	78666
	LPT	81846	91105
	MWKR	80406	80260
	FIFO	78876	83865
	RS_b	78966	81195
	RS_w	93366	91836
	RS_a	83795	85211,1
EA		74892	

Z výsledků je patrná kombinatorická složitost výroby TOS VARNSDORF a.s. díky rozdílnosti hodnot účelové funkce makespan, kde rozvrhy generované pomocí CA a RS se liší o cca 15% (RS_b – nejlepší hodnota makespan, RS_w nejhoší a RS_a průměrná). Rozdíl mezi nejhorším (CA-SPT aktivní rozvrhy) a nejlepším (EA) výsledkem makespan je jako v případě 2/3 modelu cca 34 000 minut, což odpovídá cca 23 pracovním dnům. Při porovnání s nejlepším výsledkem generovaným pomocí CA a výsledkem daným EA, dochází díky EA k úspoře necelých tří dní.

Pokud se porovná hodnota makespan mezi používaným způsobem ručního rozvrhování na lehké kovovýrobě (FIFO v kombinaci s rozvrhy bez zpoždění) s hodnotou generovanou pomocí EA, úspora činí cca 6 pracovních dní resp. je dosažena cca 14% úsporu času.

Ze srovnání mezi zákaznickým termínem (5.1.2012) a termíny odvedení zakázek dle rozvrhu daným EA (příloha č.2) lze usuzovat, že tento rozvrh lze považovat za reálný. Navržený termín odvedení byl stanoven za 2 měsíce od uvolnění zakázek do výroby s tím, že se předpokládá maximální doba odvedení zakázek 3 měsíce. Maximální zpoždění oproti stanovenému termínu je 20 pracovních dní, což odpovídá průběžné době výroby rozvrhovaných hlav. Ke zpoždění došlo u 14-ti dílců z 84 s tím, že průměrné zpoždění v případě překročení termínu je necelých 10 dní.

Posledním z výstupů pro dílenského plánovače je přehled využití středisek (příloha č.3), u kterého je sledována funkčnost nastavení směnnosti, využití časového fondu (zpracování operací, seřizování) a také blokace zakázek. Tou je míněn stav, kdy je požadavek na zpracování více zakázek najednou na jednom stroji, tzn. jedna zakázka je zpracovávána a ostatní musí čekat (jsou blokovány). Druhým případem blokace je, že zakázka musí čekat z důvodu přestávky, či v případě, kdy je stroj mimo směnu. Tato analýza může napomoci s rozhodováním o nastavení směnnosti stroje.

V případě rozvrhu EA je nejvytíženější pracoviště ručních zámečnických operací, které má využití svého pracovního fondu 76%. Na tomto pracovišti jsou také nejvíce blokovány zakázky – v 38% pracovního fondu některá ze zakázek čeká před tímto pracovištěm. Toto pracoviště lze tedy v případě tohoto modelu považovat za úzké.

7.1.5 Navržení opatření vzhledem k průběhu testování navržené metodiky

V této kapitole je provedeno zhodnocení, zda aplikovat pokročilé metody rozvrhování na modelu výrobního systému TOS VARNSDORF a.s. Vychází se přitom z provedeného testování navržené metodiky.

Hlavním prvkem umožňujícím ověření metodiky byla dlouholetá zkušenost využívání systému APS, jehož zavedení předcházely kroky podobné krokům potřebným k zavedení pokročilých heuristických metod rozvrhování. Jako zásadní při ověřování metodiky lze pokládat schopnost navrhnout realistické cíle a tyto cíle kvantifikovat (minimalizace fronty práce na pracovištích lehké kovovýroby – účelová funkce makespan).

Analýza současného stavu poukázala na některé nedostatky (např. neexistenci kusových a dávkových časů u operace kontrola), které se však podařilo v tomto kroku odstranit (pomocí kvalifikovaného odhadu). Dávkové operace, jako např. rozvrhování technologií žíhání, nitridace atd., je však nutné i nadále rozvrhovat lhůtově.

Problematika rozvrhování těchto operací je však složitá a nelze jí pokrýt klasickými rozvrhovacími metodami pro rozvrhování výrobních dávek do omezených kapacit. Jedná se o situace, kdy dochází k rozpadu klasických výrobních dávek vlivem požadavku technologie (např. naplnění žíhací pece, náběhová teplotní křivka zahřívání s ohledem na skladbu výrobků v peci atd.). Vzhledem k častému výskytu těchto technologií lze tedy doporučit tuto problematiku zkoumat hlouběji (např. v rámci navazující vědecké činnosti).

Prvkem ovlivňujícím realnost poskytovaného rozvrhu jsou také transportní a manipulační časy, které jsou v této práci navrženy na základě vzdáleností jednotlivých pracovišť a expertním odhadem provedeným obsluhou manipulačních prostředků. Tento odhad lze nahradit detailní časovou analýzou, při které je vhodné reflektovat zakázkový typ výroby (trasy resp. délka manipulace je závislá na několika faktorech (např. na rozměru, hmotnosti a trajektorii výrobním systémem)).

Další doporučení vyplývá z analýzy současného stavu a to z revize lhůt (uvolnění materiálu do výroby a termín dodání na montáž) poskytovaných stávajícím systémem APS, který však u nekritických pracovišť nerespektuje omezení vyrábět pouze jednu výrobní dávku v jeden čas na jednom stroji (plánuje se do neomezených kapacit). Tento problém lze odstranit právě aplikováním rozvrhovacího nástroje s funkcionalitou prezentovanou v této práci.

Z navazujících průběžných a finálních testů vyplývá, že v některých případech může dojít k přesunu úzkého místa dle charakteru vyráběných zakázek. Charakterem zakázek je pak myšleno množství vyráběných kusů či jejich trajektorie výrobním systémem. Tyto dva parametry ovlivňují výskyt rozvrhovacího konfliktu resp. situace, kdy je třeba rozhodnout, které zakázce se dá priorita. Toto rozhodnutí lze založit jednak na výsledku naplnění cílů daných účelovou funkcí (např. makespan), tak i pomocí dalších měřitelných ukazatelů (počet a délka zpoždění zakázek vzhledem k zákaznickému termínu, vytížení strojů atd.).

Provedené testy také prokázaly vhodnost nasazení heuristických metod na daném modelu. Přestože model lehké kovovýroby v TOS VARNSDORF a.s. není tak kombinatoricky složitý jako teoretické etalonové problémy FJSSP a JSSP, lze u něj dosáhnout zlepšení hodnoty účelové funkce ať již pomocí CA či EA.

Na základě výsledků testování navržené metodiky je možné doporučit následující kroky:

1. Revize dávkových a kusových časů v technologickém postupu u nevýrobních operací.
2. Revize způsobu stanovování lhůt náběhu zakázek do výroby a lhůt dodání hotových výrobků na finální montáž.
3. Detailní studium problematiky rozvrhování operací u technologií zpracování více různých výrobků najednou (nitridace, žihání a černění).
4. Implementace nástroje pro rozvrhování výroby pomocí pokročilých heuristických metod.

7.2 MZ Liberec a.s.

Společnost MZ Liberec a.s. se zabývá od roku 1992 výrobou příslušenství pro rozvod technických plynů, které nejčastěji dodává do zdravotnických zařízení. Hlavními vyráběnými komponenty jsou:

- Stropní stativy.
- Zdrojové mosty.
- Osvětlovací instalační rampy.
- Příslušenství ke stropním stativům, zdrojovým mostům a osvětlovacím instalačním lampám.
- Zdroje a rozvody medicínských plynů.

Společnost má v současné době cca 100-150 zaměstnanců a její roční obrát činí cca 250 miliónů (rok 2012). Dle těchto měřítek lze považovat tento podnik za malý až střední.

7.2.1 Definování požadavku na plán – rozvrh

Hlavním požadavkem společnosti MZ Liberec a.s. v oblasti plánování a rozvrhování je zmenšení celkového zpoždění odvedení zakázek dle požadovaného termínu zákazníka. Tento požadavek vychází ze způsobu plánování výrobních kapacit, který se provádí pomocí kvalifikovaného dohadu průběžné doby zpracování výrobních zakázek a kooperace, která je jedním z hlavních omezení v plánování a řízení výroby.

Mezi hlavní omezení výroby v MZ Liberec a.s. patří:

- Termín uvolnění materiálu do výroby – je dán požadavkem zákazníka a termínem dodání materiálu dodavatelem. Sortiment se vyrábí ve dvou režimech. Prvním z nich je výroba standardních dílů, které společnost vyrábí na sklad (zhromadněná výroba). Druhým typem jsou výrobky odrážející specifické požadavky zákazníka tzn., vyrábí se zakázkově.
- Strojní zařízení – toto omezení obsahuje jak obráběcí (soustruhy, frézky), tak i tvářecí stroje (lisy). Mezi důležitá patří rovněž pracoviště pájení, svařování a montáže, která kompletují vyráběné díly do finálních výrobků (zvláště v případě mostů, ramp a stativů).
- Směnný systém – v případě MZ Liberec a.s. se používá jednotný 8-mi hodinový směnný systém.
- Délky trvání výrobních dávek a trajektorie dílů výrobním systémem – je dáno technologickým postupem. Délky trvání výrobních dávek odráží plánovací režim (zakázkově či na sklad).
- Manipulace a transport – toto omezení je v případě MZ Liberec a.s. jedním ze zásadních vzhledem k tomu, že je výroba rozdělena na dvě vzdálená střediska (Liberec a Rudník) a navíc je pomocí tohoto omezení plánovaná častá kooperace.
- Poruchy, zmetky a údržba strojů – toto omezení je vázáno na seřizovací a kusové časy v poměru daném historickou zkušeností a kvalifikovaným odhadem.

7.2.2 Stanovení měřitelných ukazatelů

V této fázi se kvantifikují definované problémy a omezení tak, aby bylo možné analyzovat reálná zlepšení.

Vzhledem ke snaze minimalizovat zpoždění odvedení zakázek dle definovaných termínů zákazníkovi je stanovena jako primární účelová funkce **zpoždění všech zakázek**. Tato funkce počítá s opožděnými termíny a nebere v potaz zakázky odvedené dříve, tak jak je to obvyklé v řadě firem uplatňujících JIT. Zakázky, které jsou připraveny před stanoveným zákaznickým termínem, se často penalizují ve snaze minimalizovat náklady spojené se skladováním, což společnost vzhledem k velikosti a počtu skladovaných výrobků nepovažuje oproti riziku nedodání zakázek včas za zásadní.

Pro porovnání kombinatorické složitosti v prvních fázích tvorby modelu je však po dohodě s pracovníky společnosti použita i účelová funkce **makespan**.

Nalezená omezení se pak dále kvantifikují:

- Termín uvolnění materiálu do výroby – termíny se stanovují s přesností na dny v případě zákaznický orientovaných výrobků. Značná část výroby se však plánuje na sklad pomocí objednacích hladin, pomocí kterých se eliminují výkyvy ve výrobě. Pro jednotnost je v rámci rozvrhování použit princip shodný se zákaznickými typy výrobku.
- Strojní zařízení – vzhledem k úrovni plánování (ruční plánování – viz 7.2.3 Analýza současného stavu) je odvozena přesnost plánování od délky operací dle technologického postupu na sekundy.
- Směnný systém – je využit 2 směnný 8-mi hodinový systém s přesností vycházející z plánování strojních zařízení.
- Doba zpracování výrobních dávek a trajektorie dílů výrobním systémem – stejně jako v případě testování modelů TOS VARNSDORF a.s. jsou délky trvání výrobních dávek závislé na době seřízení (rozlišení v sekundách), době výroby jednoho kusu (sekundy) a počtu kusů ve výrobní dávce. Trajektorie zakázek výrobním systémem je stanovena na základě technologického postupu s rozlišováním jednotlivých strojů.
- Manipulace a transport – délka transportu je definovaná časem (ne vzdáleností) potřebným pro převezení materiálu mezi stroji či vstupním a výstupním skladem.
- Poruchy, zmetky a údržba strojů – není uvažováno jako primární omezení. Toto omezení je již zahrnuto v technologickém postupu dle procentuální zmetkovitosti.

7.2.3 Analýza současného stavu

Na základě definovaných cílů a stanovených měřitelných ukazatelů je provedena analýza současného stavu v následujících krocích:

- 1) Existující plánovací metody – V současné době je ve společnosti MZ Liberec a.s. používán informační systém (IS) Helios, který v tomto případě využívá plánovací jádro MRP I - plánování nákupu materiálu a výrobní lhůtové plánování do neomezených kapacit. Samotný plán je tedy tvořen ručně v těchto bodech:
 - I. Po obdržení poptávky od zákazníka je stanoven termín odvedení na základě stavu zásob na skladě, dodacích lhůt dodavatelů materiálu a odhadu průběžné doby výroby.
 - II. Na základě potvrzených objednávek zákazníka se vystaví výrobní příkazy na zákaznické díly. Výrobní příkaz mimo lhůt pro odvedení zakázky, kusového a dávkového času obsahuje informaci, kdo za danou zakázku odpovídá (nejčastěji mistr). Odpovědný pracovník pak má na starosti průchod zakázky výrobním systémem.
 - III. Odpovědný pracovník na základě kusovníkového rozpadu plánovaných zakázek tvoří výdejky materiálu na jednotlivé komponenty ve spolupráci s nákupem v případě, že se jedná o položky vstupující z nákupu přímo na finální montáž.

IV. Výrobky vyráběné na sklad se plánují podobným způsobem s tím rozdílem, že pracovník, který je odpovědný za dostupnost dílu neváže materiálovou potřebu pouze na jednu zakázku, ale na základě úrovně objednacích hladin a kvalifikovaného odhadu podpořeného spotřebou daného dílu určuje termínovou potřebu tohoto dílu.

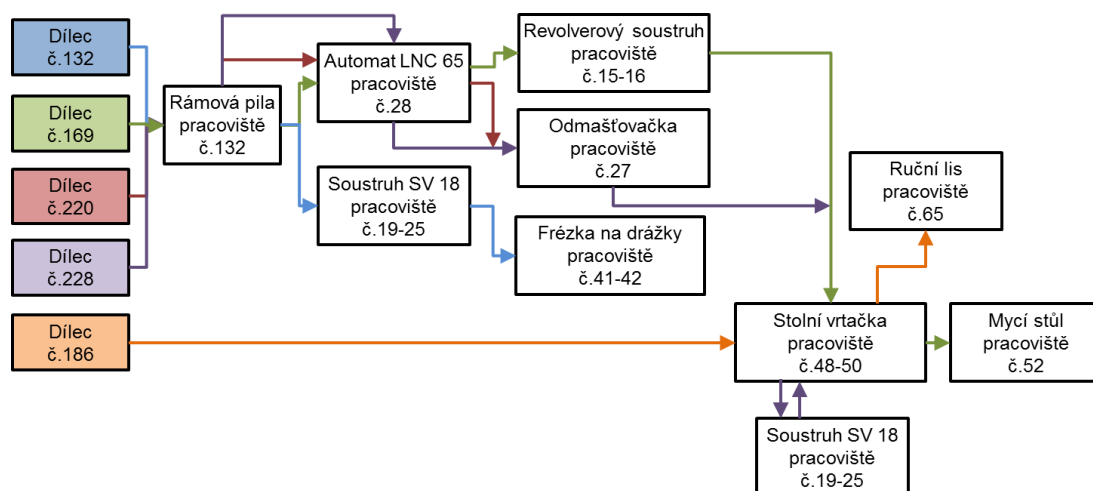
2) Časová náročnost plánování a rozvrhování – liší se výrobek od výrobku v závislosti na jeho druhu (výroba na zakázku a na sklad) a složitosti. Dlouhodobý plán výroby vychází nejen z aktuálních zakázek, ale také z odhadu založeného na předchozích prodejkách. Operativní plán výroby se tvoří ručně a reaguje na momentální změny. Tyto změny regulují jak mistři, tak tzv. přípravař výroby. Po konzultaci s pracovníky odpovědnými za plánování jsou stanoveny tři kategorie dostupného času pro použití rozvrhovací metody:

- 5 minutový interval na přeplánování výroby v průběhu směny.
- 10 minutový interval na přeplánování mezi směnami.
- 15 minutový interval na přeplánování před náběhem první směny (viz omezení směnnost bod č. 6).

3) Způsob odvádění hotové výroby – hotová výroba se odvádí průběžně ve skladu rozpracované výroby či v expedici v závislosti na tom, zda se jedná o výrobek na sklad nebo na zakázku.

4) Výběr hlavních představitelů – s ohledem na široké spektrum vyráběných dílců jsou zvoleny všechny výrobky s plánovaným náběhem výroby v posledním lednovém týdnu roku 2013. Model je doplněn o výrobky plánované na sklad s ohledem na odvolávky na materiál ze zmiňovaného plánovacího horizontu.

5) Druh výroby – na základě analýzy technologických postupů výrobků plánovaných ve zmiňovaném plánovacím horizontu je pro rozvrhování určen pro reprezentaci výrobního systému model FJSSP. Zvláštní pozornost je kladena na možnost vyrábět určité typy dílců na velkém množství strojů (flexibilita modelu). Typickým příkladem jsou soustružnické operace, které lze provádět až na 7 zařízeních. Trajektorie zakázek má pak charakter zakázkové výroby, kde se řada operací vrací na pracoviště, kde již byla zpracována a zároveň má řada dílců odlišný technologický postup (z pohledu průchodu zakázek přes stroje viz obr. 7-6).



Obr. 7-6: Příklad trajektorie výrobním systémem MZ Liberec a.s.

6) Omezení výroby

- Strojní zařízení - model výrobního systému vycházející z technologického postupu, který reflektuje výroby uvedené v úvodu kapitoly 7.2):
 - 76 strojních a ručních pracovišť obsahujících technologie dělení, obrábění, lisování, pájení, mytí, odmašťování, kontroly atd.
 - 25 pracovišť pracujících v režimu kooperace.
- Směnnost - pracoviště pracují v 8-mi hodinovém směnném systému s 2 směny, kde první směna začíná v 6:00.
- Transport – délka transportu je určena vzdáleností a způsobem dopravy mezi pracovišti. Pomocí transportu se také zohledňuje výroba v rámci kooperace.

7) Množství zakázek, operací a délka trvání operací

- Množství zakázek v modelu odráží poptávku z období 26-31.1 2013. Celkový počet výrobních dávek vstupujících do modelu je 229.
- Délka trvání výroby jednoho komponentu na jednom pracovišti má značný rozsah a pohybuje se v řádech sekund až po 30 minut.
- Doba seřízení stroje na výrobní dávku může trvat v případě zakázkových dílců až cca 120 minut. Velikosti výrobních dávek odrážejí poptávku z již zmiňovaného období. Tyto díly jsou vyráběny v dávkách po jednotkách či desítkách kusů. Drobné díly vyráběné na sklad se vyrábí ve výrobních dávkách po tisících.

8) Konzistence dat

- Identifikace zakázky – výrobní příkazy resp. jejich průvodky mají jedinečnou identifikaci pro každou z výrobních zakázek v případě zákaznických dílců. Pokud se výrobní příkaz týká výroby na sklad, řídí její identifikaci odpovědný pracovník pomocí výrobního konta. V případě dílů podsestav obsahuje průvodka informaci o nadřazené zakázce. Vzhledem k ručnímu plánování obsahuje průvodka identifikaci příslušného technologického postupu, kde jsou zaznamenány výrobní a seřizovací časy. Pro potřeby rozvrhování v navržené aplikaci dostává každý díl sestavy jednoznačnou identifikaci danou pořadovým číslem.

- Termíny uvolnění zakázek – vychází ze zákaznického termínu a sumy délky trvání operací dle technologického postupu. Nejsou zde tedy zahrnuty manipulační časy. Termín zadání do výroby je součástí průvodky na výrobní příkaz.
- Termíny odvedení zakázek – je stanoven na termín dle dohody mezi zákazníkem a společností MZ Liberec.
- Množství (požadavek a výrobní dávka) – odráží požadavek zákazníka (pokud se jedná o zákaznické dílce). V případě výroby na sklad se stanovuje ekonomická dávka dle úrovně objednacích hladin.
- Identifikace zdrojů – každý stroj má jedinečnou identifikaci včetně záměnných pracovišť. Pro potřeby rozvrhovacího modelu je každému stroji přiřazeno pořadové číslo.
- Časy kusové, seřizovací a transportní – kusové a seřizovací časy vychází z technologického postupu z informačního systému. Pro potřeby rozvrhování je provedena revize těchto časů na základě upozornění pracovníků technologie. Pro operace seřízení pak ve většině případů není nutný pro seřízení stroje vyráběný díl. Transportní časy mezi stroji a středisky jak v místě realizace Liberec, tak Rudník vychází z kvalifikovaného dohadu. Pomocí transportních operací je řešena i problematika kooperace. Délka trvání doručení kooperovaných dílů se pohybuje mezi 1-7 dny v závislosti na kapacitní disponibilitě kooperující společnosti. To znamená, že stejný typ výrobku může být doručen v tomto časovém rozsahu. Z tohoto důvodu je v případě kooperace stanovena u všech dílů 7 denní lhůta.

7.2.4 Testování modelu pomocí navržených rozvrhovacích metod

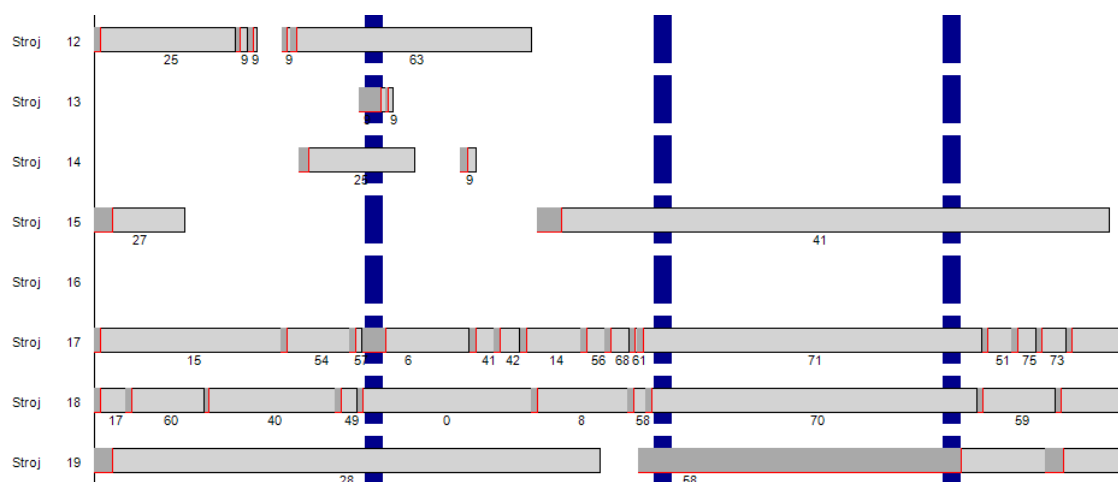
Dle stanovené metodiky je prvním krokem testování logiky stanovených omezení. V tomto případě se jedná zvláště o omezení seřizování, směnnosti a transportu. Na rozdíl od modelu TOS VARNSDORF a.s. je vybrána náhodně 1/3 výrobků dle uvedeného plánovacího horizontu. Tento 1/3 model obsahuje 76 zakázek, na kterých je provedeno 300 operací.

V první fázi je provedena ruční kontrola začátku a konce operací vybraného rozvrhu (viz. tab. 7-6) ve formě fronty práce.

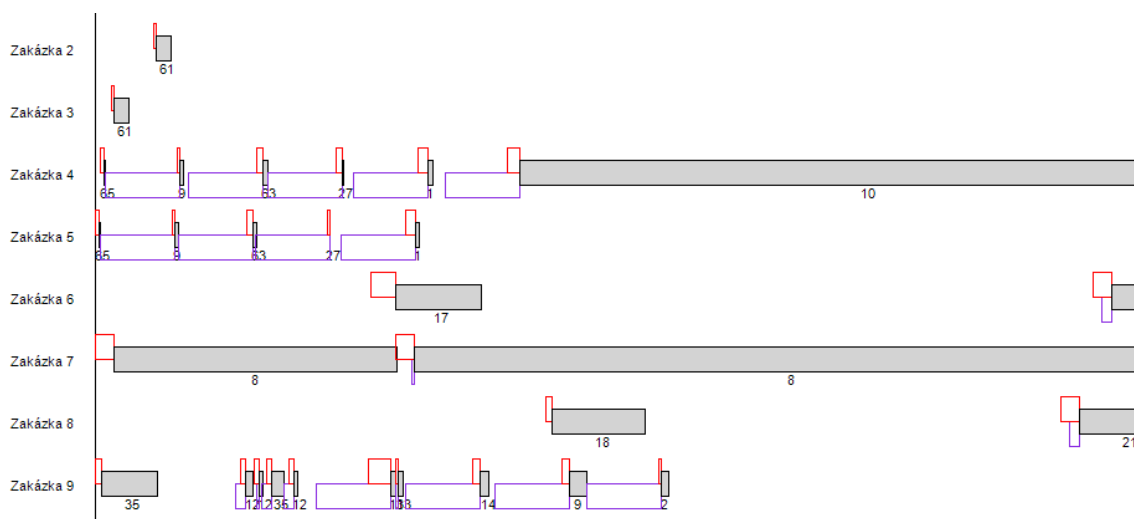
Tab. 7-6: Náhled kontroly 1/3 modelu MZ Liberec a.s.

Číslo operace	Číslo stroje	Start operace relativně [min]	Konec operace relativně [min]	Sekvence celková	Procesní čas [min]	Start operace absolutně [min]	Konec operace absolutně [min]	Seřizování [min]	Transport [min]
0	17	0	260	0	3330	21.1.2013 6:00	21.1.2013 10:20	10	0
1	28	496	1776	67	3070	21.1.2013 14:16	22.1.2013 11:36	120	15
2	27	1880	1921	104	2300	22.1.2013 13:20	22.1.2013 14:01	40	120
3	49	1936	2071	106	2265	22.1.2013 14:16	22.1.2013 16:31	10	15
4	33	2086	3166	109	2130	22.1.2013 16:46	23.1.2013 10:46	20	15
5	9	3281	3591	146	1560	23.1.2013 12:41	23.1.2013 17:51	5	120
6	65	3711	4501	157	1280	23.1.2013 19:51	24.1.2013 9:01	30	120
7	81	9541	10041	208	1000	27.1.2013 21:01	28.1.2013 5:21	0	5040
8	80	15081	15581	237	500	31.1.2013 17:21	1.2.2013 1:41	0	5040

V rámci tohoto kroku je provedena stejně jako u modelu TOS VARNSDORF a.s. namátková vizuální kontrola směnnosti (modré bloky) a seřizování (šedivé bloky s červeným rámováním) pomocí Ganttova diagramu z pohledu přes stroje (obr. 7-7 na ose y jsou stroje -) a kontrola transportu (modře rámovaný bílý blok) zobrazením přes zakázky (obr. 7-8 na ose y jsou zakázky).



Obr. 7-7: Ganttův diagram - ukázka pohledu přes stroje MZ Liberec a.s.



Obr. 7-8: Ganttův diagram - ukázka pohledu přes zakázky

Následně je provedena analýza časové náročnosti a odhad kombinatorické složitosti modelu. V této fázi je přes definovaný cíl minimalizace zpoždění zakázek použita účelová funkce makespan vzhledem k tomu, že 1/3 modelu neobsahuje termíny zákazníka a termíny uvolnění materiálu do výroby (resp. tyto termíny jsou pro všechny zakázky stejné).

Testování je provedeno pomocí konstruktivního algoritmu (CA) spolu s prioritními pravidly SPT, LPT, MWKR, FIFO a RS (deset rozvrhů). Rozvrhování s použitím aktivních rozvrhů a rozvrhů bez zpoždění trvá 36 s (na PC sestavách viz kapitola 6 - úvod) při 300 operacích.

Ačkoli je pro prvotní testování zvolen pouze 1/3 model, je patrná velká kombinatorická složitost (tab. 7-7). To je patrně způsobeno velkým množstvím operací (300). Maximální rozdíl mezi nejlepším (aktivní rozvrh RS_b) a nejhorším výsledkem (rozvrh bez zpoždění LPT) účelové funkce makespan je až cca 10 000 minut (cca 7 pracovních dní). Výsledky generované pomocí RS mají jak v případě aktivních rozvrhů,

tak rozvrhů bez zpoždění, lepší hodnotu makespan. Je tedy pravděpodobné, že EA v rámci AO dosáhne lepších výsledků než rozvrhy generované pomocí CA.

Tab. 7-7: Hodnoty účelové funkce makespan generované pomocí CA 1/3 modelu MZ Liberec a.s.

Prioritní pravidla	Aktivní rozvrhy [min]	Rozvrhy bez zpoždění [min]
SPT	42970	42821
LPT	46231	49389
MWKR	41911	45335
FIFO	41471	39506
RS_b	39130	39272
RS_w	41471	47671
RS_a	40161	44072,8

V dalším kroku je provedeno testování na 2/3 modelu (152 zakázek). Nejprve je testován model bez vazby na termíny uvolňování materiálu pro odhad kombinatorické složitosti. V této fázi je již model optimalizován pomocí kaskádové automatické optimalizace (AO tab. 7-8) s časem pro optimalizaci 5, 10 a 15 minut. Tento čas vychází z požadavků pracovníků společnosti (viz kapitola 7.2.3 - časová náročnost plánování a rozvrhování).

Model obsahující 2/3 zakázek, jejichž celkový počet výrobních dávek nutných ke zpracování je 600, kaskádový algoritmus testuje metodou CA a EA, přičemž metodu LS vynechává vzhledem k předpokládané vysoké časové náročnosti (odhad na základě trvání CA).

Tab. 7-8: Hodnoty účelové funkce makespan generované pomocí AO 2/3 modelu MZ Liberec a.s.

Metoda	Prioritní pravidla	Aktivní rozvrhy [min]	Rozvrhy bez zpoždění [min]
CA	SPT	246184	246184
	LPT	242606	206615
	MWKR	243478	248388
	FIFO	244654	243678
	RS_b	241177	198164
	RS_w	243495	249207
	RS_a	242808	223630
EA 5 [min]		196289	
EA 10 [min]		195251	
EA 15 [min]		194889	

Z výsledků je patrné, že přidáním 1/3 objemu zakázek do modelu se značně zvýšila hodnota funkce makespan. To je způsobeno zejména přidáním zakázek, které obsahují operace prováděné v kooperaci s délkou trvání (lhůtou dodání kooperace) 7 dní.

Rozdíly mezi prioritními pravidly jsou značné, což naznačuje stále vysokou míru kombinatorické složitosti. Porovnáním nejobvykleji používaného pravidla FIFO v kombinaci s rozvrhy bez zpoždění a nejlepšího výsledku generovaného náhodně pomocí stejného druhu rozvrhu lze dojít až k úspoře cca 31 dní a procentuálnímu zlepšení cca 21%.

Evoluční algoritmus v rámci AO dosáhl nejlepších výsledků v závislosti na poskytnutém čase. Zlepšení oproti nejlepšímu výsledku daného pomocí CA při investici 15 minut do optimalizace činí cca 2 pracovní dny tzn. dochází k cca 2% úspoře času.

V dalším kroku jsou do modelu zavedeny termíny uvolňování materiálu do výroby pro porovnání reálnosti generovaných rozvrhů a lze tedy zkontrolovat reálnost nastavení termínů odvádění zákazníkovi.

Pro porovnání zákaznických termínů s termíny odvedení hotové výroby je použito pravidlo FIFO s rozvrhy bez zpoždění, vzhledem k tomu, že je to pravidlo, které se nejčastěji používá ve společnosti. Z testu modelu s vlivem uvolnění materiálu do výroby vyplývá nevhodné nastavení termínů (viz příloha č. 4) a velikosti výrobních dávek výrobků vyráběných na sklad. To je patrné zejména z velkého množství zakázek (24 zakázek z 152) s nedodrženým termínem, u nichž zpoždění dosahuje více než 100 dní.

Termíny a výrobní dávky všech zakázek jsou zkorigovány na základě množství obvyklého pro doplňování stavu zásob a je doplněna poslední třetina zakázek pro testování finálního modelu. Model je testován nejprve pomocí účelové funkce makespan s dostupným časem pro optimalizaci 15 minut (viz tab. 7-9).

Tab. 7-9: Hodnoty účelové funkce makespan generované pomocí AO plného modelu MZ Liberec a.s.

Metoda	Prioritní pravidla	Aktivní rozvrhy [min]	Rozvrhy bez zpoždění [min]
CA	SPT	57130	55684
	LPT	59020	62414
	MWKR	55724	63574
	FIFO	59019	59998
	RS_b	52139	64804
	RS_w	55685	70565
	RS_a	54185	67128,7
EA [min]		51930	

Výsledky ukazují, že po korekci termínů náběhu a velikosti výrobních dávek, zvláště v případech výroby na sklad, se velikost účelové funkce makespan rapidně zmenšila. Úspora díky použití EA v porovnání s pravidlem FIFO s rozvrhy bez zpoždění činí více než 5 pracovních dní, respektive došlo k cca 14% úspoře času.

Vzhledem k požadavku společnosti MZ Liberec a.s. na minimalizaci celkového zpoždění všech zakázek je dále model testován na základě této účelové funkce. Optimalizace probíhá jako případ 2/3 modelu v dostupných časech na optimalizaci 5, 10 a 15 minut (tab. 7-10).

Tab. 7-10: Hodnoty účelové funkce makespan generované pomocí AO plného modelu MZ Liberec a.s.

Metoda	Prioritní pravidla	Aktivní rozvrhy [min]	Rozvrhy bez zpoždění [min]
CA	SPT	549486	396195
	LPT	435955	1597071
	MWKR	504731	1774077
	FIFO	387718	1104151
	RS_b	428785	948764
	RS_w	483500	1273940
	RS_a	455320	1044510
EA 5 [min]		405389	
EA 10 [min]		377979	
EA 15 [min]		339234	

Z výsledků optimalizace je patrné, že při dostupném času optimalizace 5 minut EA nenalezne lepší řešení než CA. To je způsobené velikostí modelu (229 zakázek obsahujících 934 výrobních dávek) resp. délkou trvání tvorby rozvrhů pomocí CA (4:30 minuty). Při optimalizaci v délce 10 a 15 minut je pak dosažen lepší výsledek než pomocí CA. V případě 15 minutové optimalizace je zkráceno zpoždění všech zakázek o cca 163 pracovních dní v porovnání s pravidlem FIFO a rozvrhů bez zpoždění. Pokud se porovnají výsledky z EA a nejlepším výsledkem CA (FIFO s aktivními rozvrhy) je dosaženo úspory 33 pracovních dní, tzn. výsledek se zlepší o cca 13%.

Celková hodnota zpoždění všech zakázek se jeví na první pohled při daném plánovacím horizontu jako nereálná. Je však třeba si uvědomit, že se jedná o zpoždění generované 229 zakázkami. Průměrné zpoždění u nejlepšího výsledku je tedy cca 1 pracovní den (viz příloha č. 5a a 5b) a dosažená úspora vůči nejlepšímu výsledku CA činí průměrně necelé 4 hodiny. Celkový počet zpožděných zakázek je 77 z 229, tzn. v cca 33% procentech zakázek dojde ke zpoždění, přičemž nejdelší zpoždění čítá 21 pracovních dní. Na druhou stranu použitím obvyklého systému řízení (FIFO a rozvrhy bez zpoždění) dojde k trojnásobnému zpoždění všech zakázek.

Důležitým výstupem pro dílenského plánovače je také přehled využití středisek, u kterého je sledována funkčnost nastavení směnnosti, využití časového fondu (zpracování operací, seřizování) a také blokace zakázek.

V případě rozvrhu EA je nejvytíženější pracoviště (viz příloha č. 6a a 6b) automat LNC 65 (66%). Toto pracoviště lze pokládat za úzké místo. Dalších 8 pracovišť má vytížení vyšší než 30% (druhé nejvyšší vytížení je 50%) a procento blokování zakázek je méně než 20%.

7.2.5 Navržení opatření vzhledem k výsledkům testů

V této kapitole je provedeno zhodnocení, zda aplikovat pokročilé metody rozvrhování na modelu výrobního systému MZ Liberec a.s. Vychází se přitom z provedeného testování navržené metodiky.

V prvním a druhém kroku je definován a kvantifikován cíl zmenšit celkové zpoždění všech zakázek spolu s identifikováním hlavních omezení a způsobu jejich rozvrhování.

Analýza současného stavu poukazuje zejména na stávající způsob plánování a rozvrhování, který lze hodnotit jako jednu z největších překážek pro zavedení pokročilých heuristických metod.

Společnost sice využívá informační systém pro plánování výroby, jeho hlavní náplní je však spíše ekonomické řízení společnosti, než řízení výroby. Samotné plánování uvnitř tohoto systému je na úrovni MRP I, a tedy plánuje do neomezených kapacit pomocí lhůtového plánu.

Jednou z hlavních překážek pro zavedení rozvrhovacího nástroje je koncept řízení výroby, kde část výroby se vyrábí na sklad (řízení pomocí objednacích hladin) a část zakázkově. Hlavní odpovědnost leží na zaměstnancích ručících za úroveň zásob na skladě a zaměstnancích majících na starosti zakázkové díly.

Tento problém lze omezit systémem s plánovacím jádrem MRP II obsahujícím nejen podporu při rozpadu kusovníků, ale také volbu velikosti výrobních dávek vzhledem k zhromadněné výrobě a zákaznickým dílcům. Důležitým modulem takového informačního systému je pak dílenské řízení výroby a zvláště pak systém odvádění hotové výroby, který umožňuje efektivní využití výrobních kapacit a připraví společnost na řízení pomocí rozvrhovacích metod.

Přesto lze konstatovat, že je společnost připravena na zavedení pokročilejšího nástroje plánování, ať již MRP II či rozvrhovacího nástroje. To lze usoudit z existující datové základny technologických postupů, na základě kterých je vytvořen model pro další testování. Hlavním omezením tvorby modelu vycházejícím z analýzy současného stavu je systém kooperace a řízení dodavatelských vztahů. To vyplývá z velkého rozsahu dodacích lhůt, kde u jednoho typu výrobku se může doba dodání pohybovat mezi 3-7 dny v průběhu zpracování zakázky. Tento rozsah zásadně ovlivňuje pořadí zakázek zpracovávaných na následujících operacích a má negativní efekt na zvolenou účelovou funkci zpoždění všech zakázek. Tuto situaci však nelze řešit informačním systémem. Zde je třeba uplatnit některou z lehkých manažerských technik.

Testování 1/3 a 2/3 modelu poukázalo na omezení vyplývající ze současného stavu plánování výroby. Jedná se zejména o velikost výrobních dávek dílů vyráběných na sklad a nerealisticky stanovených termínů odvedení zákazníkovi.

Výsledky tohoto testování dále potvrdily předpoklad velké kombinatorické složitosti modelu a zároveň poukázaly na velkou časovou náročnost rozvrhování vycházející z počtu rozvrhovaných zakázek a jejich výrobních dávek.

Po úpravě termínů náběhu do výroby, termínu odvedení zákazníkovi a velikosti výrobních dávek je na plném modelu provedeno testování (pomocí účelové funkce makespan), jak tato změna ovlivnila kombinatorickou složitost modelu a tím vhodnost nasazení pokročilých rozvrhovacích metod. Porovnáním výsledků generovaných pomocí CA a EA uvnitř AO, kde EA dosáhlo 21% úspory makespan lze konstatovat vhodnost použití EA na daný model.

Vhodnost použití AO potvrdila optimalizace modelu dle účelové funkce zpoždění všech zakázek, kde je v tomto případě dosaženo 13% redukce zpoždění generované pomocí samotného CA.

Na základě průběhu ověření navržené metodiky lze stanovit následující doporučení:

- Změna způsobu plánování – implementace plánovacího nástroje využívající metody MRP II.
- Zavedení systému řízení dodavatelských vztahů, zejména pak standardizace dodacích lhůt.
- Zavedení systému automatické identifikace a systému dílenského řízení výroby pro odvádění hotové výroby.
- Opětovné posouzení vhodnosti implementace pokročilých rozvrhovacích metod na základě navržené metodiky.
- Pokud výsledky testování prokážou podobné úspory jako tomu je při testování současného modelu, pak lze doporučit zavedení pokročilých rozvrhovacích metod prezentovaných v této práci.

7.3 Dílčí závěr č.5 – Zhodnocení ověření metodiky využívání heuristických algoritmů v rozvrhování výroby

Pro ověření navržené metodiky jsou použity modely dvou výrobních systémů - společnosti TOS VARNSDORF a.s. a MZ Liberec a.s.

Každý z těchto modelů popisuje typickou strojírenskou výrobu s prvky zakázkové výroby. Oba modely, jak TOS VARNSDORF a.s. respektive jeho popisovaná část - lehká kovovýroba, tak MZ Liberec a.s., splňují parametry (počet zaměstnanců a obrát) malého až středního podniku.

Současný způsob plánování a rozvrhování se v těchto společnostech liší. Společnost TOS VARNSDORF a.s. má dlouholeté zkušenosti s využíváním systému pro pokročilé plánování a rozvrhování využívající principy plánování na základě úzkých míst. Pomocí tohoto APS se řídí typicky zakázková výroba.

Společnost MZ Liberec a.s. pro plánování a řízení výroby používá informační systém využívající principy MRP I a plánuje tedy do neomezených kapacit pomocí lhůtového plánu. Detailní rozvrhování se dělí dle způsobu zásobování finální montáže na díly vyráběné na sklad nebo zakázkově, tzn. na požadavek zákazníka. Pro oba typy výrobku jsou použity na úrovni dílenského plánu principy ručního rozvrhování.

Analýza současného stavu mimo běžných omezení (stroje, termíny náběhu výroby atd.) poukazuje na zásadní omezení, které je v těchto modelech velice obtížné rozvrhovat pomocí klasických metod zakázkového rozvrhování, a proto je na ně použit systém lhůtového plánování.

V případě TOS VARNSDORF a.s. je to omezení technologiemi chemicko-fyzikálních operací jako je žíhání nebo černění, kde se výrobní dávky rozpadají na jednotlivé díly dle specifik dané technologie (např. velikost pracovního prostoru žíhací pece).

Ve společnosti MZ Liberec a.s. se jedná o podobné technologie resp. operace, které jsou však poptávány v rámci externí kooperace. Hlavním omezením je v tomto případě široký rozsah dodacích lhůt (3-7dní), které se mění v průběhu zpracování zakázek.

V obou dvou případech se tyto operace rozvrhují lhůtově s délkou trvání stanovenou na základě historické maximální hodnoty.

Analýza současného stavu dále poukazuje na některé dílčí nedostatky. Jedná se zejména o neexistující kusové (procesní) a seřizovací časy operací (typicky operace kontroly) či nerealisticky nastavené termíny náběhu výroby a termínu odvedení zákazníkovi, způsobené zejména odhadem průběžné doby výroby nerespektující vedlejší časy (manipulace a transport).

Testování modelů výše zmiňovaných společností prokázalo vhodnost zavedení pokročilých optimalizačních metod navržených v této práci. Doporučení zavést tyto metody se opírá zejména o celkové úspory vzniklé implementací EA do AO. V případě TOS VARNSDORF a.s. je to 14% snížení potřebného času na odvedení všech zakázek (makespan), u MZ Liberec a.s. 13% redukce zpoždění všech zakázek.

V obou případech lze zavést zmiňované metody za určitých omezujících podmínek. Jedná se zejména o revize technologických postupů, termínů náběhu a odvedení hotové výroby. V případě MZ Liberec a.s. se navíc doporučuje standardizace dodavatelských vztahů (zejména dodacích lhůt) a pozvolné zavedení systému obsahující MRP II před zavedením samotných rozvrhovacích metod.

8 ZÁVĚREČNÉ SHRNUTÍ DISERTAČNÍ PRÁCE

Tato práce se zabývá rozvrhováním strojírenské výroby a její optimalizací pomocí pokročilých heuristických metod. Hlavní motivací této práce je zlepšit stav v oblasti plánování a rozvrhování výroby v SME, kdy na jedné straně se stále využívají principy ručního rozvrhování, které jsou přímo závislé na zkušenosti dílenského plánovače a často neposkytují kvalitní (z pohledu účelové funkce) rozvrhy, a na straně druhé existují pokročilé heuristické algoritmy, které se přes jejich schopnost nalézat velmi kvalitní rozvrhy neimplementují vzhledem k jejich relativní složitosti. V této závěrečné části jsou shrnuty cíle disertační práce, její teoretické a praktické přínosy a naznačeny směry dalšího výzkumu.

8.1 Shrnutí cílů disertační práce

Hlavní cíl této práce bylo **navrhnout metodiku využívání heuristických algoritmů v rozvrhování výroby v SME společně s optimalizačním nástrojem rozvrhování výrobních operací pro potřeby dílenského resp. operativního rozvrhování**. Těchto cílů bylo dosaženo následujícími kroky:

- 1) Analýza problémů rozvrhování výroby, popis používaných výrobních modelů (kap.3).

Cílem tohoto bodu bylo analyzovat stávající teoretické modely rozvrhování výroby a vybrat takový, který by nejlépe popisoval často složitou trajektorii výrobku výrobním systémem.

Analýza se zabývala základními třemi modely a to otevřeným, proudovým a sekvenčním modelem. Z analýzy těchto modelů vyplynulo:

- Otevřený rozvrhovací problém (OSSP) řeší v praxi pouze segmenty proudových či sekvenčních modelů.
- Proudový model (FSSP) je úspěšně řešen pomocí metod Lean.
- Sekvenční problém (JSSP) má největší možnost uplatnění principů rozvrhování ve strojírenské výrobě díky možnosti popsat jakoukoli trajektorii výrobku výrobním systémem.

Pro další práci byl proto vybrán model sekvenčního rozvrhovacího problému (JSSP) a jeho flexibilní modifikace (FJSSP), která je v omezené míře schopna řešit otevřený rozvrhovací problém.

- 2) Studium možností generování výrobních rozvrhů pro daný model včetně metod pro jejich optimalizaci. (kap. 4).

Cílem bylo nalezení metod využitelných pro návrh softwarové aplikace (příloha č.7). V tomto bodu byly analyzovány metody (algoritmy) rozvrhování a způsob jejich optimalizace. S ohledem na předpokládanou časovou náročnost optimalizace a předpokládanou kvalitu poskytovaných řešení (rozvrhů) byly pro optimalizační nástroj zvoleny následující algoritmy:

- Konstruktivní algoritmus pro tvorbu rychlých rozvrhů pomocí prioritních pravidel.
- Metoda lokálního prohledávání optimalizující řešení z konstruktivního algoritmu za předpokladu vyšší časové dotace než je tomu u konstruktivních algoritmů.

- Evoluční algoritmus využívající poznatky z oblasti memetických algoritmů a evolučních strategií pro nalezení optimálních či optimu blízkých řešení při dostatečné časové rezervě.
- 3) Navržení metodiky využití optimalizačních metod v oblasti rozvrhování výrobních zakázek (kap. 5).

Cílem bylo navržení metodiky využití rozvrhovacího nástroje v oblasti strojírenské výroby. Tato metodika navrhuje postup pro posouzení vhodnosti implementace a samotnou implementaci rozvrhovacích algoritmů na reálné modely strojírenské výroby. Zohledňuje přitom:

- Schopnost nadefinovat a kvantifikovat reálné cíle implementace pokročilých rozvrhovacích nástrojů.
- Schopnost kvantifikovat omezení modelu.
- Analýzu současného stavu plánovacích metod využívaných ve společnosti a jejich časovou náročnost. Zásadním bodem této analýzy je tvorba modelu výroby vzhledem k množství zakázek v plánovacím horizontu, charakteru prováděných operací a konzistenci dostupných dat o použitých omezeních (výrobní a nevýrobní časy, směnnost, kooperace atd.).
- Výsledky testování modelů, kde je brán zřetel jak na schopnost modelu popsat reálný výrobní systém, tak i na poskytnutí reálných rozvrhů ve stanovený čas.

V tomto kroku jsou testovány metody tvorby rozvrhu a jejich optimalizace. Výsledkem je stanovení podmínek implementace těchto metod.

- 4) Navržení, implementace a otestování optimalizačních algoritmů ve vztahu ke známým optimálním řešením a časové náročnosti optimalizace. (kap. 6).

Cíl této části je naplněn v následujících krocích:

- Návrh konstruktivního algoritmu (CA), který je schopný rozvrhovat JSSP a FJSSP problémy. Je posouzena schopnost tohoto algoritmu dosahovat pomocí prioritních pravidel optimálních řešení s důrazem na potřebný výpočetní čas. Z testování tohoto algoritmu na 20-ti teoretických problémech vyplývá, že pro různé modely jsou vhodná různá prioritní pravidla. Nejlepších výsledků dosahuje prioritní pravidlo nejvíce práce zbývá (MWKR) společně s pravidlem první příchozí je prvním zpracovaným (FIFO). Velmi dobrých výsledků dosahuje také pravidlo náhodného výběru (RS) v případě, že je těchto rozvrhů generováno větší množství než jeden (bylo testováno 10 rozvrhů).
- Návrh principu lokálního prohledávání (LS) jedním prohozením, který při testování na vybraných 20 teoretických problémech dosahuje lepších výsledků než je tomu u CA. Nevýhodou tohoto algoritmu je delší čas pro optimalizaci a jeho časté uvíznutí v lokálním extrému. V případě, že je tento extrém počátečním řešením daným CA znamená, že LS není schopné nalézt lepší řešení. Nevýhodou tohoto algoritmu je také generování nereálných rozvrhů, což znehodnocuje investovaný optimalizační čas.
- Návrh evolučního algoritmu (EA). V rámci vývoje tohoto algoritmu byla testována různá nastavení optimalizačních operátorů selekce, křížení a mutace na dvou nejobtížnějších modelech. Toto testování mělo za úkol posoudit vliv těchto parametrů na dosaženou hodnotu účelové funkce a potřebný optimalizační čas. Největší vliv na tyto hodnoty má operátor selekce a mutace. S ohledem na

hodnoty účelové funkce a optimalizační čas je vybrána elitářská selekce s kontrolou klonů, mutace uplatněná na 10% populace a 10% pravděpodobnost křížení. Toto nastavení je testováno na celkem 20-ti teoretických modelech JSSP a FJSSP. EA dosahuje známých optimálních či optimu blízkých hodnot účelové funkce a je zvolen pro kaskádovou automatickou optimalizaci jako nejpokročilejší nástroj rozvrhování.

- Návrh kaskádové automatické optimalizace využívající CA, LS a EA. Tato optimalizace na základě zadané účelové funkce a dostupného času pro optimalizaci aplikuje výše zmíněné metody CA, LS a EA. EA v kaskádové optimalizaci využívá adaptivního řízení velikosti populace, která zásadním způsobem ovlivňuje délku optimalizace. Adaptivní řízení optimalizace pomocí EA dosahuje kvalitativně podobných výsledků jako je tomu u ručního nastavování. Tím je splněn cíl vytvoření takového principu optimalizace, který bude v praxi jednoduchým způsobem aplikovatelný.
- 5) Ověření navržené metodiky a optimalizačních algoritmů na reálných modelech strojírenské výroby pomocí softwarového nástroje (viz příloha č. 7).

Navržená metodika (kap. 5) je ověřena na modelech reálné výroby TOS VARNSDORF a.s. a MZ Liberec a.s. V obou případech bylo dosaženo pomocí automatické kaskádové optimalizace zlepšení účelových funkcí:

- TOS VARNSDORF a.s. – makespan (čas dokončení všech zakázek) byl snížen o 14% v porovnání s FIFO a rozvrhy bez zpoždění, které se používají v současné době ve společnosti TOS.
- MZ Liberec a.s. – zpoždění všech zakázek je sníženo o 13% v porovnání s FIFO a rozvrhy bez zpoždění, které se používají při rozvrhování ve společnosti MZ.

Na základě ověřování metodiky je doporučena implementace pokročilého nástroje (automatické kaskádové optimalizace) pro rozvrhování s následujícími omezujícími podmínkami:

TOS VARNSDORF a.s.:

1. Revize dávkových a kusových časů v technologickém postupu u nevýrobních operací.
2. Revize způsobu stanovování lhůt náběhu zakázek do výroby a lhůt dodání hotových výrobků na finální montáž.
3. Detailní studium problematiky rozvrhování operací u technologií zpracování více různých výrobků najednou (nitridace, žíhání a černění).

MZ Liberec a.s.:

1. Změna způsobu plánování – implementace plánovacího nástroje využívající metody MRP II.
2. Zavedení systému řízení dodavatelských vztahů, zejména pak standardizace dodacích lhůt.
3. Zavedení systému automatické identifikace a systému dílenského řízení výroby pro odvádění hotové výroby.
4. Posouzení vhodnosti implementace pokročilých rozvrhovacích metod na základě navržené metodiky.

8.2 Teoretické přínosy disertační práce

Teoretické přínosy této disertační práce jsou:

- Navržení lokálního prohledávání s jedním prohozením operací.
- Navržení evolučního algoritmu využívajícího opravu chromosomu, kontroly klonu a hybridizaci lokálním prohledáváním.
- Ověření navržených algoritmů na známých teoretických problémech, což vedlo k nalezení významných operátorů optimalizace pomocí EA a jejich nastavení.
- Navržení a ověření kaskádové automatické optimalizace na teoretických problémech.
- Navržení metodiky využívání heuristických algoritmů v rozvrhování výroby v SME.

8.3 Praktické přínosy disertační práce

Praktické přínosy této disertační práce jsou:

- Ověření metodiky využívání heuristických algoritmů v rozvrhování výroby v SME na modelech reálné výroby.
- Vytvoření programové aplikace pro rozvrhování výroby pomocí navržených algoritmů.
- Ověření kaskádové automatické optimalizace na praktických příkladech.
- Doporučení pro testované podniky, které se týká oblasti plánování a rozvrhování výroby.

8.4 Směry dalšího výzkumu

Hlavním směrem dalšího výzkumu vyplývajícím z průběhu ověřování metodiky a tvorby rozvrhovacích algoritmů je studium rozvrhování dávkových technologií, které jsou omezeny dalšími faktory, jako jsou pracovní prostor stroje a nastavení regulátorů těchto zařízení. Jedná se o regulátory, jako například teplotní křivka u žíhacích pecí či chemické složení lázně u povrchových úprav. Předpokládá se, že tento problém lze řešit pomocí optimalizačních algoritmů řešících problém batohu s více omezeními.

Tuto problematiku je tedy vhodné řešit pomocí technik rozvrhování výroby s ohledem na velikost a skladbu výrobních dávek, které jsou omezeny výše popsanými regulátory.

Nedílnou součástí výzkumu by měla být řešena problematika:

- Tvorby sekvence vstupu výrobních dávek řazených dle druhu výrobku do zmíněných technologií.
- Vytvoření nových výrobních dávek dle požadavku technologie.
- Spojování/rozdělování výrobních dávek dle typu výrobku.

Dalším směrem výzkumu bude také hodnocení rizik při rozhodování o přijetí nenadálých zakázek (tzv. vlomů).

Literatura

- [1] WEAVER, P. A BRIEF HISTORY OF SCHEDULING: - BACK TO THE FUTURE -. In: [online]. [cit. 2012-01-31]. Dostupné z: http://www.mosaicprojects.com.au/PDF_Papers/P042_History%20of%20Scheduling.pdf
- [2] SCHERER, Eric . Results of the ERP user satisfaction survey : Conclusion for the real-world scheduling. In[online]. [s.l.] : 7th International Workshop on Human Factors in Planning, Scheduling and Control in Manufacturing, 2005 [cit. 2011-10-07]. Dostupné z WWW: <http://www.hops-research.org/DocAttach/Presentation/M10_62_108200521387xzhhtjyqncedrxz.pdf>
- [3] HÁDEK, L . *Organizace a řízení výroby II* [online]. 1. Ostrava : Bussines school ostrava plc, 2006 [cit. 2011-10-21]. Dostupné z WWW : <http://www.ronge.cz/dokumenty/skripta_VS/M_4_organizace_a_rizeni_vyroby_II.pdf>.
- [4] BRAHMANKAR, R . *SAP Production Planning* SlideShare [online]. 2009 [cit. 2011-10-07].. Dostupné z WWW: <http://www.slideshare.net/rupesh_b4u2/sap-production-planning>.
- [5] RICE, T. Production Scheduling by Spreadsheet [online]. 2011 [cit. 2011-10-07]. Production-Scheduling. Dostupné z WWW: <<http://production-scheduling.com/welcome-to-the-new-production-scheduling-com/>>.
- [6] KUČERÁK, D. MRP III: IPA Slovník. In: [online]. 2008 [cit. 2013-05-09]. Dostupné z: <http://www.ipaczech.cz/cz/ipa-slovník/mrp-iii>
- [7] IVER, L K. Advanced planning and scheduling systems in manufacturing planning and control processes [online]. Göteborg, Sweden : Chalmers Reproservice, 2009. 51 s. Rigorózní práce. Chalmers University of Technology. Dostupné z WWW: <<http://www.iei.liu.se/prodek/forskning/iscaps/filarkiv/1.204241/LineaKjellsdotterIvertKlar.pdf>>. ISSN 1654-973
- [8] Production Planning and Detailed Scheduling : e-optimization community [online]. 1999 [cit. 2011-10-07]. SAP Advanced Planner&Optimizer. Dostupné z WWW: <http://e-opt.informs.org/resources/uploads/PP_DS.pdf>.
- [9] ŠILHÁNEK, P; a ZAJÍČEK, V. Jak uřídit výrobu v dnešní turbulentní době. IT Systems. Září 2011, 9/2011, s. 32-34. ISSN 1802-002X.
- [10] HESTON, T. The fabricator [online]. August 4, 2010 [cit. 2011-10-05]. For CEOs Article. Dostupné z WWW: <http://www.thefabricator.com/article/forceos/can-lean-manufacturing-work-in-the-job-shop>

- [11] The Problem with Push Production Control Systems [online]. 2002 [cit. 2011-10-05]. Kanban Systems and Visual Controls. Dostupné z WWW: <<http://www.ct-yankee.com/lean/kanban.html>>.
- [12] SURI, R. A Lean Strategy for Job Shops POLCA is a system for high-variety or custom-engineered products. GEAR TECHNOLOGY, 2005, VOL 22; NUMB 6, strany 26-27, ISSN 44743-6858
- [13] GRAVEL, M a PRICE W. L. Using the Kanban in a job shop environment. International Journal of Production Research, Volume 26, Issue 6 June 1988, strany 1105 – 1118.
- [14] BASL, J; MAJER, P; ŠMÍRA, M. Teorie omezení v podnikové praxi. dotisk 2005. Havlíčkův Brod : Grada Publishing, 2003. 213 s. ISBN 80-247-0613-X.
- [15] ADAMS, J, BALAS, E, ZAWACK D: The shifting bottleneck procedure for job shop scheduling, Management Science, v.34 n.3, p.391-401, March 1988
- [16] DERMIKOL, E, MEHTA, S., UZSOY, R, A. Computational Study of Shifting Bottleneck Procedures for Shop Scheduling Problems, Journal of Heuristics, v.3 n.2, p.111-137, Winter 1997
- [17] MONCH, L, ZIMMERMAN J. Simulation-based assessment of machine criticality measures for a shifting bottleneck scheduling approach in complex manufacturing systems. Computers in Industry Volume 58 , Issue 7 (September 2007) Pages 644-655 ISSN:0166-3615
- [18] DRIESSEL, R, et al. A parallel shifting bottleneck heuristic for scheduling complex job shops: Architecture and performance assessment. In Automation Science and Engineering (CASE), 2010. Toronto, ON : [s.n.], 2010. s. 81 - 86 . Dostupné z WWW: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5584066&tag=1#Index+Terms>. ISBN 978-1-4244-5447-1
- [19] PETTIT, R. G. Job shop type production scheduling by simulation. In Proceedings of the Second Conference on Applications of Simulations (New York, New York, United States, December 02 - 04, 1968). Winter Simulation Conference. Winter Simulation Conference, 254-259.
- [20] QINGSONG L. a DAN Q. Simulation and Optimization of Discrete Customized Job-Shop Scheduling, Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference on , vol.2, no., pp.213-215, 10-11 Oct. 2009 doi: 10.1109/ICICTA.2009.289 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5288138&isnumber=5287724>
- [21] MANLIG, F. Optimierung von Fertigungsprozessen mit Rechnersimulation In: Forschungsergebnisberichte 2001 der Professur für Produktionsautomatisierung Steuerungstechnik. [Forschungsbericht] TU Dresden, PAS, 2001

- [22] OTTJES J.A. a VEEKE.H.P.M. Production Scheduling of Complex Jobs with Simulation. Proceedings of the Business and Industry Simulation Symposium (ASTC 1999). April 2000. Washington D.C. [SCS]. ISBN 1-56555-199-0
- [23] ALPAY, Ş. Agent Based Dynamic Job Shop Simulation System. Lecture Notes in Computer Science. Springer. Berlin / Heidelberg Volume 4570/2007 ISBN 978-3-540-73322-5 Pages 364-373.
- [24] MANLIG, F., VAVRUŠKA, J., DUŠÁKOVÁ, A: Podpora rozvrhování výroby pomocí počítačové simulace. Strojírenská technologie, 2008, zvláštní číslo,s. 157-160. ISSN 1211-4162
- [25] BRUCKER, P. Scheduling Algorithms. 5th edition. [s.l.] : Springer, 2007. 383 s. ISBN 978-3540695158, 354069515X.
- [26] KOLISH, R. a PADMAN, R., An integrated survey of deterministic project scheduling, Omega, Volume 29, Issue 3, June 2001, Pages 249-272, ISSN 0305-0483, 10.1016/S0305-0483(00)00046-3.
- [27] BRUCKER, P., DREXL A., MOHRING R., NEUMAN K., PESCH, E. Resource-constrained project scheduling: Notation, classification, models, and methods, European Journal of Operational Research, Volume 112, Issue 1, 1 January 1999, Pages 3-41, ISSN 0377-2217, 10.1016/S0377-2217(98)00204-5.
- [28] GHALLAB M., NAU D., TRAVERSO P. Automated Planning: Theory and Practice, Morgan Kaufmann, 2004
- [29] PINEDO M.L, Scheduling: Theory, Algorithms, and Systems, Third Edition, Springer, 2008
- [30] BRASSARD, G. a BRATLEY P. Algorithmics: Theory and Practice. 1. USA: Prentice-Hall, 1988. ISBN 0130232432.
- [31] BELLMORE, M. a NEMHANS G. L. , The Traveling Salesman Problem: a Survey, Operations Research, 16(1968), 538-55
- [32] BALAS, E. a ZEMEL, E., An algorithm for large zero-one knapsack problems, Operations Research 28 (1980) 1130–1154.
- [33] LODI, A., MARTELLO, S., MONACI, M. Two-dimensional packing problems: A survey (2002) European JoUrnal of Operations Research, 141 (2), PP. 241-25
- [34] GHIANI, G. a IMPROTA, G. An efficient transformation of the generalized vehicle routing problem, Eur. J. Oper. Res., 122 (2000), pp. 11–17

- [35] JIANG, Ch. A Reliable Solver of Euclidean Traveling Salesman Problems with Microsoft Excel Add-in Tools for Small-size Systems. Journal of Software. 2010-07-01, vol. 5, issue 7, s. -. DOI: 10.4304/jsw.5.7.761-768. Dostupné z: <http://www.academypublisher.com/ojs/index.php/jsw/article/view/2677>
- [36] BURKARD, R.E., Selected topics on assignment problems. Discrete Applied Mathematics 123 (1–3), 257–302. 2002.
- [37] KARP, R. M. 1986. Combinatorics, complexity, and randomness. Communications of the ACM 29, 98-109.
- [38] PELIKÁN, J.: Diskrétní modely Skripta VŠE Praha, 1999
- [39] PLESNÍK, J.: Grafové algoritmy, VEDA, Bratislava, 1983.
- [40] Evoluční výpočetní techniky: principy a aplikace. 1. české vyd. Praha: BEN, 2009, 534 s. ISBN 978-80-7300-218-3.
- [41] DEOLALIKAR, V. $P \neq NP$. [online]. 6.8.2010 [cit. 2012-01-30]. Dostupné z: <http://www.scribd.com/doc/35539144/pnp12pt>
- [42] NIELSEN, M. Polymath project: Deolalikar P vs NP . [online]. [cit. 2012-01-30]. Dostupné z: http://michaelnielsen.org/polymath1/index.php?title=Deolalikar_P_vs_NP_paper
- [43] REHMEYER, J. Crowdsourcing peer review: A claimed proof that $P \neq NP$ spurs a massive collaborative research effort. In: ScienceNews [online]. 9.9.2010 [cit. 2012-01-30]. Dostupné z: http://www.sciencenews.org/index/generic/activity/view/id/63252/title/Crowdsourcing_peer_review
- [44] GONZALEZ, T. a SAHNI S., Open shop scheduling to minimize finish time, J. Assoc. Comput. Mach. 23 (1976), pp. 665-679.
- [45] LAWLER, E.L., LENSTRA, J.K., RINNOOY KAN, A.H.G., Minimizing maximum lateness in a two-machine open shop, Math. Oper. Res., 6(1):153-158, 1981.
- [46] CHENG, T.C.E. a SHAKHLEVICH, N., Two-machine open shop problem with controllable processing times, 2006.
- [47] KUBALE M. a NADOLSKI A., Chromatic scheduling in a cyclic open shop, European Journal of Operational Research, Volume 164, Issue 3, 1 August 2005, Pages 585-591, ISSN 0377-2217, 10.1016/j.ejor.2003.06.047.
- [48] LIAW, C.F. An efficient tabu search approach for the two-machine preemptive open shop scheduling; computers & Operations Research 30, 2081-2095. (2003)

- [49] LAWER , E.L., LENSTRA, J.K., A.H.G., RINNOOY KAN D.B. S. Sequencing and Scheduling: Algorithms and Complexity, in: S.C.Gtaves, A.H.G.Rinnooy Kan, P.H.Zipkin(Eds.), Handbooks in Operation Research and Management Science, v. 4, Logistics of Production and Inventory, North_holland, Amsterrdem, 445-522, 1993.
- [50] LING, S. a CHENG X., Complexity Results for Flow-shop Scheduling Problems with Transportation Delays and a Single Robot, Journal of Applied Mathematics & Bioinformatics, vol.1, no.1, 2011, 135-142 ISSN: 1792-6602 (print), 1792-6939 (online)
- [51] RUIZ, R a VAZQUEZ-RODRIGUEZ J. A. The hybrid flow shop scheduling problem European Journal of Operational Research 205 (2010) 1–18, DOI: <http://dx.doi.org/10.1016/j.ejor.2009.09.24>
- [52] DJELLAB, H., a DJELLAB, K. Preemptive hybrid flowshop scheduling problem of interval orders, European Journal of Operational Research 137 (1) (2002) 37–
- [53] GUPTA J.N.D., Two-stage, hybrid flow shop scheduling problem, Journal of the Operational Research Society 39 (4) (1988) 359–364.
- [54] LIU, C.-Y., CHANG, S.-C., Scheduling flexible flow shops with sequencedependent setup effects, IEEE Transactions on Robotics and Automation 16 (4) (2000) 408–419.
- [55] JIN Z.H., OHNO K., ITO T., S.E. Scheduling hybrid flowshops in printed circuit board assembly lines, Production and Operations Management 11 (2) (2002) 216–230.
- [56] SHERALI, H.D., SARIN S.C., KODIALAM M.S., Models and algorithms for a two-stage production process, Production Planning and Control 1 (1) (1990) 27–39.
- [57] GRABOWSKI, J. PEMPERA, J., Sequencing of jobs in some production system, European Journal of Operational Research 125 (3) (2000) 535–550.
- [58] AGNETIS, A. et al., Scheduling of flexible flow lines in an automobile assembly plant, European Journal of Operational Research 97 (2) (1997) 348–362.
- [59] KURZ,M.E. ASKIN, R.G. Scheduling flexible flow lines with sequence dependent setup times, European Journal of Operational Research 159 (1) (2004) 66–82.
- [60] T.J. SAWIK, Mixed integer programming for scheduling flexible flow lines with limited intermediate buffers, Mathematical and Computer Modelling 31 (13)(2000) 39–52.

- [61] R. TAVAKKOLI-MOGHADDAM, N. SAFAEI, F. SASSANI, A memetic algorithm for the flexible flow line scheduling problem with processor blocking, *Computers. and Operations Research* 36 (2) (2009) 402–414.
- [62] JAIN, A.S., MEERAN, S, Deterministic job-shop scheduling: Past, present and future, *European Journal of Operational Research*, Volume 113, Issue 2, March 1999, Pages 390-434, ISSN 0377-2217, 10.1016/S0377-2217(98)00113-1
- [63] TIMKOVSKY, VG. The complexity of unit-time job-shop scheduling', 15th International Symposium on Mathematical Programming, The University of Michigan, United States, 1st January 1994-31st December 1994
- [64] VADIM G. a TIMKOVSKY, A polynomial-time algorithm for the two-machine unit-time release-date job-shop schedule-length problem, *Discrete Applied Mathematics*, Volume 77, Issue 2, 8 August 1997, Pages 185-200, ISSN 0166-218X, 10.1016/S0166-218X(96)00106-0.
- [65] BRUCKER, P. a KNUST. *Complex Scheduling*. 2. vyd. London: Springer, 2011. ISBN 978-3642239281
- [66] IDSIA Simulation and Optimisation of Transport Systems: Istituto Dalle Molle di Studi sull'Intelligenza Artificiale. [online]. [cit. 2012-03-25]. Dostupné z: <http://www.idsia.ch/intermodal/>
- [67] RUBIYAH, Y. MARZUKI, K. TAY, CH. S. Solving industrial based job-shop scheduling problem by distributed micro-genetic algorithm with local search. In *Proceedings of the 14th international conference on Knowledge-based and intelligent information and engineering systems: Part I (KES'10)*, Rossitza Setchi, Ivan Jordanov, Robert J. Howlett, and Lakhmi C. Jain (Eds.). Springer-Verlag, Berlin, Heidelberg, 69-79.
- [68] PÉREZ-VÁZQUEZ, M.E. GENTO-MUNICIO, A.M. LOURENÇO, H.R. Solving a concrete sleepers production scheduling by genetic algorithms, *European Journal of Operational Research*, Volume 179, Issue 3, 16 June 2007, Pages 605-620, ISSN 0377-2217, 10.1016/j.ejor.2005.03.070.
- [69] RUI, Z. a CHENG, W. A hybrid local search algorithm for scheduling real-world job shops with batch-wise pending due dates, *Engineering Applications of Artificial Intelligence*, Volume 25, Issue 2, March 2012, Pages 209-221, ISSN 0952-1976, 10.1016/j.engappai.2011.04.014.
- [70] HANSMANN, K.-W. a, HOECK, M. (1997). Production control of a flexible manufacturing system in a job shop environment. *International Transactions in Operational Research*, 4(5/6), 341–351.

- [71] TOWNSEND, W.B. Artificial Intelligence Techniques for Industrial Applications in Job Shop Scheduling, NAVAL POSTGRADUATE SCHOOL MONTEREY CA. Defense Technical Information Center <http://books.google.cz/books?id=zdajcQAACAAJ> 1983
- [72] HONGYAN, X. a HONG, H. Research on job-shop scheduling problem based on Self-Adaptation Genetic Algorithm," Logistics Systems and Intelligent Management, 2010 International Conference on , vol.2, no., pp. 940-943, 9-10 Jan. 2010 doi: 10.1109/ICLSIM.2010.5461042
- [73] STRUSEVICH, V.A.: Two machine super-shop scheduling problem. Journal of the Operational Research society, 42(6):479-492, 1991
- [74] MASUDE, T., ISHII, H, NISHIDA, T., The mixed shop scheduling problem, Discrete Applied Mathematics, Volume 11, Issue 2, June 1985, 175-186, ISSN 0166-218X, DOI: 10.1016/S0166-218X(85)80007-X
- [75] SHAKHLEVICH, N.V., SOTSKOV, Yu.N. WERNER, F. Shop scheduling problem with fixed and non fixed machine orders of jobs. Annals of Operation Research, 92:281-304, 1999.
- [76] SHAKHLEVICH, N.V, SOTSKOV, Yu.N. WERNER, F. Scheduling two jobs with fixed and non fixed routes Computing, 52(1)17-30, 1994
- [77] SHAKHLEVICH, N.V., SOTSKOV Yu.N. and WERNER, F.: Mixed shop scheduling problems. Project report INTAS (96-0820), ISTS(B-1104-98)
- [78] ZOBOLAS, G., I., TARANTILIS, C., D., IOANNOU, G.: Exact, Heuristics and Metaheuristic Algorithms for Solving Shop Scheduling Problems. Studies in Computational Intelligence, 2008, Volume 128/2008, 1-40, DOI: 10.1007/978-3-540-78985-7_1
- [79] LUI, S., Q., ONG, H., L., Metaheuristic for the mixed shop scheduling problem. Asia-Pacific Journal of Operational Research (APJOR) 2004 DOI: 10.1142/S021759504000072
- [80] KLEEMAN, G. a LAMONT, B. Scheduling of Flow-Shop, Job-Shop, and Combined Scheduling Problems using MOEAs with Fixed and Variable Length Chromosomes. EVOLUTIONARY SCHEDULING: Studies in Computational Intelligence 2007
- [81] SAMPELS M., BLUM C., MASTROLILLI M., ROSSI-DORIA O.. Metaheuristics for Group Shop Scheduling. In J. J. Merelo Guerv' o et al., editor, Proceedings of PPSN-VII, Seventh International Conference on Parallel Problem Solving from Nature, number 2439 in Lecture Notes in Computer Science, pages 631–640, Berlin, Germany, 2002. Springer Verlag.
- [82] DVOŘÁK, J., ŠEDA, M., VLÁČIL, T. Rozvrhování zakázkové výroby. In XXI. seminář ASŘ '98 "Počítače v měření, diagnostice a řízení. ČR,

Ostrava- Poruba: Vysoká škola báňská - Technická univerzita Ostrava, 1998. s. 1 (s.)ISBN: 80-7078-559- 4.

- [83] JAMES M. WILSON, Gantt charts: A centenary appreciation, European Journal of Operational Research, Volume 149, Issue 2, 1 September 2003, Pages 430-437, ISSN 0377-2217, 10.1016/S0377-2217(02)00769-5.
(<http://www.sciencedirect.com/science/article/pii/S0377221702007695>)
- [84] DILEEPAN a P. SEN T., Bicriterion jobshop scheduling with total flowtime and sum of squared lateness, Engineering Costs and Production Economics, Volume 21, Issue 3, July 1991, Pages 295-299, ISSN 0167-188X, 10.1016/0167-188X(91)90008-P.
- [85] LING-HUEY, S., PEI-CHANN, CH. LEE, E.S. A heuristic for scheduling general job shops to minimize maximum lateness, Mathematical and Computer Modelling, Volume 27, Issue 1, January 1998, Pages 1-15, ISSN 0895-7177, 10.1016/S0895-7177(97)00250-1
- [86] MATTFELD, D.a C.,BIERWIRTH, CH. An efficient genetic algorithm for job shop scheduling with tardiness objectives, European Journal of Operational Research, Volume 155, Issue 3, 16 June 2004, Pages 616-630, ISSN 0377-2217, 10.1016/S0377-2217(03)00016-X.
- [87] ASANO M. a OHTA H., A heuristic for job shop scheduling to minimize total weighted tardiness, Computers & Industrial Engineering, Volume 42, Issues 2–4, 11 April 2002, Pages 137-147, ISSN 0360-8352, 10.1016/S0360-8352(02)00019-0
- [88] MOSLEHI, G. MIRZAEI, M. VASEI, M. MODARRES, M., AZARON A. Two-machine flow shop scheduling to minimize the sum of maximum earliness and tardiness, International Journal of Production Economics, Volume 122, Issue 2, December 2009, Pages 763-773, ISSN 0925-5273, 10.1016/j.ijpe.2009.07.003
- [89] KHER HV, MALHOTRA MK, Acquiring and operationalizing worker flexibility in dual resource constrained job shops with worker transfer delays and learning losses, Omega, Volume 22, Issue 5, September 1994, Pages 521-533, ISSN 0305-0483, 10.1016/0305-0483(94)90032-9.
- [90] FATTAHI P., FALLAHI A., Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability, CIRP Journal of Manufacturing Science and Technology, Volume 2, Issue 2, 2010, Pages 114-123, ISSN 1755-5817, 10.1016/j.cirpj.2009.10.001.
- [91] HILL TH. W. Jr., Colored Petri Net Modeling of Metal Distribution in a Job Shop Iron Foundry: Improving Flow in a Pull System with Perishable In-Process Inventory, Journal of Manufacturing Processes, Volume 5, Issue 1, 2003, Pages 66-77, ISSN 1526-6125, 10.1016/S1526-6125(03)70042-5.

- [92] KACEM, I. HAMMADI S., BORNE P., Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic, *Mathematics and Computers in Simulation*, Volume 60, Issues 3–5, 30 September 2002, Pages 245-276, ISSN 0378-4754, 10.1016/S0378-4754(02)00019-8
- [93] BAKER K.R Introduction to Sequencing and Scheduling John Wiley & Sons Inc. ISBN-13: 978-0471045557 (November 1974)
- [94] JOHNSON, Donald B. (1977), "Efficient algorithms for shortest paths in sparse networks", *Journal of the ACM* 24 (1): 1–13, doi:10.1145/321992.321993
- [95] GONÇALVES, J. F., MAURICIO RESENDE G. C.. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*. 2011, vol. 17, issue 5, s. 487-525. DOI: 10.1007/s10732-010-9143-1. Dostupné z: <http://link.springer.com/10.1007/s10732-010-9143-1>
- [96] BRUCKER P., JURISCH B., SIEVERS B., A branch and bound algorithm for the job-shop scheduling problem, *Discrete Applied Mathematics*, Volume 49, Issues 1–3, 30 March 1994, Pages 107-127, ISSN 0166-218X, 10.1016/0166-218X(94)90204-6.

(<http://www.sciencedirect.com/science/article/pii/0166218X94902046>)
- [97] KUCERA L., *Kombinatorické Algoritmy*, SNTL, Praha 1983
- [98] WANG, J., LUH, P.B., ZHAO, X. An Optimization-Based Algorithm for Job Shop Scheduling, *SADHANA, a Journal of Indian Academy of Sciences, a Special Issue on Competitive Manufacturing Systems*, Vol. 22, Part 2, April 1997, pp. 241–256.
- [99] GROMICHO A.S., VAN HOORN J., SALDANHA-DA-GAMA F., TIMMER G. T., Solving the job-shop scheduling problem optimally by dynamic programming, *Computers & Operations Research*, Volume 39, Issue 12, December 2012, Pages 2968-2977, ISSN 0305-0548, 10.1016/j.cor.2012.02.02
- [100] PANWALKAR S.S., ISKANDER W.: A survey of scheduling rules. *Operations Research*, Vol. 25, No.1, 45-60, 1977.
- [101] WERNER, F., WINKLER ,A. Insertion techniques for the heuristic solution of the job shop problem, *Discrete Applied Mathematics*, Volume 58, Issue 2, 24 March 1995, Pages 191-211, ISSN 0166-218X, 10.1016/0166-218X(93)E0127-K.
- [102] GIFFLER B., THOMPSON G.: *Algorithms for Solving Production Scheduling Problems*. *European Journal of Operational Research*, 1960, vol. 8, pp. 487-503.

- [103] GAO J, GEN M, SUN L, ZHAO X, A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems, *Computers & Industrial Engineering*, Volume 53, Issue 1, August 2007, Pages 149-162, ISSN 0360-8352, 10.1016/j.cie.2007.04.010.
- [104] [WERNER, F. WINKLER, A. Insertion techniques for the heuristic solution of the job shop problem, *Discrete Applied Mathematics*, Volume 58, Issue 2, 24 March 1995, Pages 191-211, ISSN 0166-218X, 10.1016/0166-218X(93)E0127-K.
- [105] PESEK, I., ŽEROVNIK J., Best insertion algorithm for resource-constrained project scheduling problem 11th International Conference on Operational Research KOI 2006
- [106] VAESSENS, R. J. AARTS M., LENSTRA. J.K. Job Shop Scheduling by Local Search. *INFORMS JOURNAL ON COMPUTING*. 1994, s. 302-317.
- [107] KVASNICKA, V., PELIKAN M. POSPICHAL, J. Hill Climbing with Learning - An Abstraction of Genetic Algorithm. *Neural Network World*. 1996.
- [108] DUNN, S.A. The use of genetic algorithms and stochastic hill-climbing in dynamic finite element model identification, *Computers & Structures*, Volume 66, Issue 4, 15 February 1998, Pages 489-497, ISSN 0045-7949, 10.1016/S0045-7949(97)00092-8.
- [109] OVALLE-MARTÍNEZ, F. J., GONZÁLEZ J. S. STOJMENOVIĆ, I. A parallel hill climbing algorithm for pushing dependent data in clients-providers-servers systems. *Mobile Network and Applications*. roč. 2004, č. 9.
- [110] CEYLAN, H. (2006). "Developing Combined Genetic Algorithm—Hill-Climbing Optimization Method for Area Traffic Control." *J. Transp. Eng.*, 132(8), 663–671.
- [111] AARTS, E.H.L. a P.J.M. VAN LAARHOVEN. Simulated annealing: An introduction. *Statistica Neerlandica*: Blackwell Publishing Ltd. 1989, roč. 9, č. 1, s. 31-52. DOI: 10.1111/j.1467-9574.1989.tb01245.x.
- [112] AARTS, E.H.L., VAN LAARHOVEN P.J.M., LENSTRA J. K.. Job Shop Scheduling by Simulated Annealing. *Operations Research*. 1992, roč. 40, č. 1, s. 113-125.
- [113] C. GARCÍA-MARTÍNEZ, M. LOZANO, F.J. RODRÍGUEZ-DÍAZ, A simulated annealing method based on a specialised evolutionary algorithm, *Applied Soft Computing*, Volume 12, Issue 2, February 2012, Pages 573-588, ISSN 1568-4946, 10.1016/j.asoc.2011.11.007.

- [114] REEVES, C.R. Modern Heuristic Techniques for Combinatorial problems. Oxford, Blackwell Scientific Publications. 1993
- [115] GLOVER, F. a LAGUNA, M., 1997, Tabu Search, Kluwer, Boston, MA.
- [116] NOWICKI E., SMUTNICKI C.: A fast tabu search algorithm for job shop problem. Manag. Sci., vol. 42, pp. 797–813, 1996.
- [117] HOLLAND J. H.: Adaptation in Natural and Artificial Systems. University of Michigan Press. 183 pp. 1975
- [118] KOZA J. R.. Genetic Programming: On the Programming of Computers by Means of Natural Selection. A Bradford Book. The MIT Press, Cambridge, Massachusetts, USA, 1992. ISBN: 0-2621-1170-5, 978-0-26211-170-6.
- [119] HEITKOTTER J. a BEASLEY, D. Hitch-Hiker's Guide to Evolutionary Computation: A List of Frequently Asked Questions (FAQ). ENCORE (The Evolutionary Computation REpository Network), 1998. USENET: comp.ai.genetic. On line <http://www.cse.dmu.ac.uk/~rij/gafaq/top.htm>
- [120] WEISE T, Global Optimization Algorithms, 2008, elektronická kniha, dostupná z <http://www.it-weise.de/projects/book.pdf> 10.7.2012
- [121] ULRYCH, Z. a RAŠKA, P. Modelování a simulace a DP. 1. vyd. Plzeň : SmartMotion, 2012, ISBN: 978-80-87539-15-6
- [122] FOGEL D. B., editor. Evolutionary Computation: The Fossil Record. Wiley-IEEE Press, May 1, 1998. ISBN: 978-0-78033-481-6
- [123] HOFFMEISTER, F. a SCHWEFEL H-P. Korr 2.1 – an implementation of a $(\mu + \lambda)$ -evolution strategy. Technical Report, Universität Dortmund, Fachbereich Informatik, November 1990. Interner Bericht.
- [124] SCHWEFEL H-P. Numerical Optimization of Computer Models. John Wiley and Sons Ltd, New York, NY, USA, June 17, 1981. ISBN: 0-4710-9988-0, 978-0-47109-988-8.
- [125] GEYER H., ULBIG P., and Siegfried Schulz. Encapsulated evolution strategies for the determination of group contribution model parameters in order to predict thermodynamic properties. In PPSN V: Proceedings of the 5th International Conference on Parallel Problem Solving from Nature, pages 978–987, 1998.
- [126] BOSWORTH, J., FOO, N. ZEIGLER, B. P.. Comparison of genetic algorithms with conjugate gradient methods. ORA Tech Report 00312-1-T, Computer and Communication Sciences Department, The University of Michigan, Ann Arbor, Michigan, USA, 1972.

- [127] BETHKE, A. D. Genetic algorithms as function optimizers. PhD thesis, Uni-versity of Michigan, University of Michigan, Ann Arbor, MI, USA, 1980. Order No. AAI8106101, Dissertation Abstracts International, 41(9), 3503B (University Micro-films No. 8106101).
- [128] GROSAN, C. a ABRAHAM, A. Hybrid evolutionary algorithms: Methodologies, architectures, and reviews. In Hybrid Evolutionary Algorithms, pages 1–17. Springer, 2007. doi:10.1007/978-3-540-73297-6 1.
- [129] MOSCATO, P. On evolution, search, optimization, genetic algorithms and mar-tial arts: Towards memetic algorithms. Technical Report C3P 826, Caltech Con-current Computation Program 158-79, California Institute of Technology, Pasadena,CA 91125, USA, Pasadena, CA, 1989.
- [130] Chevalier de LAMARCK, Jean-Baptiste Pierre Antoine de Monet. Philosophie zoologique – ou Exposition des consid'érations relatives `a l'histoire naturelle des Animaux; `a la diversit'é de leur organisation et des facult'es qu'ils en obtiennent; aux causes physiques qui maintiennent en eux la vie et donnent lieu aux mouvements qu'ils ex'ecutant; enfin, `a celles qui produisent les unes le sentiment, et les autres l'intelligence de ceux qui en sont dou'es. Dentu / J. B. Bailli'ere Libraire, De L'Acad'emie Royale de M'edecine, Rue de l' Ecole de M'edecine 13, Paris, France / Harvard University, 1809, 1830. ISBN: 1-4121-1646-5,978-1-41211-646-6.Online
http://www.lamarck.cnrs.fr/ice/ice_book_detail.php?lang=en&type=text&bdd=lamarck&table=ouvrages_lamarck&bookId=29<http://books.google.fr/books?id=L6qAG6ZPgj4C> 10.7.2012
- [131] SIMPSON G. G. The baldwin effect. *Evolution*, 7(2):110–117, červen 1953. ISSN: 0014-3820.
- [132] SUZUKI R. a ARITA T. Repeated occurrences of the baldwin effect can guide evolution on rugged fitness landscapes. In *Proceedings of the IEEE Symposium on Artificial Life (CI-ALife'07)*, pages 8–14. Omnipress, April 1–5, 2007. doi:10.1109/ALIFE.2007.367650. Online available at http://www.alife.cs.is.nagoya-u.ac.jp/~reiji/publications/2007_ieeealife_suzuki.pdf 10.7.2012
- [133] JERNE, N. K.. Towards a network theory of the immune system. *Annual Review of Immunology (Inst. Pasteur)*, 125(C), 373-389. (1974)
- [134] PERELSON, A. S. Immune network theory. *Immunological Review* (1989), 110, 5-36. DOI: 10.1111/j.1600-065X.1989.tb00025.x
- [135] TIMMIS, J., KNIGHT, T., de Castro, L. N., & HART, E. An overview of artificial immune systems. *Natural Computation Series*, 51-86. New York: Springer (2004).. ISBN 978-3540003588.

- [136] LAHA, D. "Heuristics and Metaheuristics for Solving Scheduling Problems." Handbook of Computational Intelligence in Manufacturing and Production Management. IGI Global, 2008. 1-18. Web. 11 Jul. 2012. doi:10.4018/978-1-59904-582-5.ch001
- [137] DENEUBOURG, J-L. PASTEELS, J-M, and VERHAEGHE, J. C.. Probabilistic behaviour in ants: a strategy of errors? Journal of Theoretical Biology, 105(2):259–271, 1983. doi:10.1016/S0022-5193(83)80007-1
- [138] STICKLAND, T. R. TOFTS, C. M. N. FRANKS, N. R. A path choice algorithm for ants. Naturwissenschaften, 79(12):567–572, December 1992. ISSN: 0028-1042
- [139] DORIGO, M. MANIEZZO, V. COLORNI, A. The Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics, 26(1):29–41, 1996. Online <ftp://iridia.ulb.ac.be/pub/mdorigo/journals/IJ.10-SMC96.pdf> 11.7.2012
- [140] ZHANG J., HU X. M., TAN X., ZHONG J. H. HUANG Q. Implementation of an ant colony optimization technique for job shop scheduling problem", Trans. Inst. Meas. Control, vol. 28, pp.93 2006 doi: 10.1191/0142331206tm165oa
- [141] EBERHART R. C. a KENNEDY J. A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science MHS'95, pages 39–43. IEEE Press, October 1995. ISBN: 0-7803-2676-8.
- [142] GAO Y. a REN Z.. Adaptive particle swarm optimization algorithm with genetic mutation operation. In Proceedings of the Third International Conference on Natural Computation (ICNC 2007), volume 2, pages 211–215, 2007. doi:10.1109/ICNC.2007.161.
- [143] ANANDARAMAN, Ch.. An improved sheep flock heredity algorithm for job shop scheduling and flow shop scheduling problems. International Journal of Industrial Engineering Computations. 2011, roč. 2, č. 4, s. 749-764. ISSN 19232926. Dostupné z: http://growingscience.com/ijiec/Vol2/IJIEC_2011_34.pdf
- [144] EBERHART R.C. a SHI Y.. Comparison between genetic algorithms and particle swarm optimization. In Proceedings of the 7th International Conference on Evolutionary Programming, pages 611–616, 1998.
- [145] RAŠKA, P. a ULRYCH, Z., "Simulation Optimization In Manufacturing Systems." Zadar : DAAAM, 2012. stránky 221-224. ISBN 978-3-901509-91-9. ISSN 2304-1382.

- [146] VOTAVA, V., ULRYCH, Z., EDL, M., KORECKÝ, M., TRKOVSKÝ, V.: Analysis and Optimization of Complex Small-lot Production in New Manufacturing Facilities Based on Discrete Simulation. In: 20th European Modeling & Simulation Symposium, Italy, 2008. str. 198-203. ISBN 978-88-903724-0-7
- [147] VAVRUŠKA, J. MANLIG, F. KOBLASA, F.: One piece flow – Caravans, deeper recognition. In: International doctoral seminar, Smolenice květen 2009, ISBN 978-80-8096-088-9
- [148] KOPEČEK, P. Optimalizace denního rozvrhu výroby. Plzeň, 2009., ISBN: 978-80-7043-844-2
- [149] FISHER H., a THOMPSON G.L.: Probabilistic learning combinations of local job-shop scheduling rules”, J.F. Muth, G.L. Thompson (eds.), Industrial Scheduling, Prentice Hall, Englewood Cliffs, New Jersey, 225-251., 1963
- [150] STORER R.H., WU S.D., VACCARI R.: New search spaces for sequencing instances with application to job shop scheduling, Management Science 38, 1495-1509., 1992
- [151] LAWRENCE S.: Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (Supplement), Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania., 1984
- [152] BRANDIMARTE, P., Routing and scheduling in a flexible job shop by tabu search, Annals of Operations Research, vol. 41, pp. 157~V183, 1993.
- [153] GEN, M., a CHENG, R.: Genetic algorithms and engineering design, Wiley series in engineering design and automation 1997 ISBN 96-8437
- [154] BEAN, J.C., Genetic Algorithms and Random Keys for Sequencing and Optimization, ORSA Journal on Computing, vol.6, no.2, Spring 1994
- [155] OKADA, I., F. ZHANG,, X. H. Y. YANG a S. FUJIMURA. A Random Key-based Genetic Algorithm Approach for Resource-constrained Project Scheduling Problem with Multiple Modes. Proceedings of the International MultiConference of Engineers and Computer Scientists 2010. 2010. ISSN 2078-0958. Dostupné z: http://www.iaeng.org/publication/IMECS2010/IMECS2010_pp106-111.pdf
- [156] YAMADA T., a NAKANO, R. Genetic algorithms for job shop scheduling problems Proceedings of modern heuristic for decision support (1997), pp. 67–81

- [157] PEREZ, E. Job Shop Scheduling Problem: Benchmark Instances. [online]. 2011, 06/2011 [cit. 2013-03-14]. Dostupné z: <http://www.eii.uva.es/elena/JSSP/InstancesJSSP.htm>
- [158] TSUNG-CHE, Ch., a HSIAO-JOU, L., A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling, *International Journal of Production Economics*, Volume 141, Issue 1, January 2013, Pages 87-98, ISSN 0925-5273, 10.1016/j.ijpe.2012.03.034. (<http://www.sciencedirect.com/science/article/pii/S0925527312001429>)
- [159] GOTSHALL,S. and RYLANDER, B., □Optimal Population Size And The Genetic Algoithm.□, *Proc On Genetic And Evolutionary Computation Conference*, 2000.
- [160] TIANSHI Ch., TANG, K. GUOLIANG, Ch, XIN Y., A large population size can be unhelpful in evolutionary algorithms, *Theoretical Computer Science*, Volume 436, 8 June 2012, Pages 54-70, ISSN 0304-3975, 10.1016/j.tcs.2011.02.016. (<http://www.sciencedirect.com/science/article/pii/S0304397511001368>)
- [161] QING-DAO-ER-JI, R. WANG, Y., A new hybrid genetic algorithm for job shop scheduling problem, *Computers & Operations Research*, Volume 39, Issue 10, October 2012, Pages 2291-2299, ISSN 0305-0548, 10.1016/j.cor.2011.12.005. (<http://www.sciencedirect.com/science/article/pii/S0305054811003601>)
- [162] F. PEZZELLA, G. MORGANTI, G. CIASCETTI, A genetic algorithm for the Flexible Job-shop Scheduling Problem, *Computers & Operations Research*, Volume 35, Issue 10, October 2008, Pages 3202-3212, ISSN 0305-0548, 10.1016/j.cor.2007.02.014.
- [163] GIBBS, M.S.; MAIER, H.R.; DANDY, G.C.; NIXON, J.B., "Minimum Number of Generations Required for Convergence of Genetic Algorithms," *Evolutionary Computation*, 2006. CEC 2006. IEEE Congress on , vol., no., pp.565,572, 0-0 0 doi: 10.1109/CEC.2006.1688360
- [164] LI, Y. CHEN, Y. A Genetic Algorithm for Job Shop Scheduling, *Journal of software*,2010,pp.269-274.
- [165] MESGHOUNI, K., HAMMADI, S. BORNE P. Evolutionary algorithm for job shop scheduling. - *Int. J. Appl. Math. Comput. Sci.*, Vol. 14, No. 1, pp. 91-103. (2004)
- [166] XU; L. SHUANG W; HUANG M., Application on job-shop scheduling with Genetic Algorithm based on the mixed strategy, *Control and Decision Conference*, 2009. CCDC '09. Chinese , vol., no., pp.2007,2009, 17-19 June 2009 doi: 10.1109/CCDC.2009.5191650

- [167] T. HU , W. BANZHAF. Evolvability and speed of evolutionary algorithms in light of recent developments in biology, *Journal of Artificial Evolution and Applications*, (2010, 1-28)
- [168] Elizabeth MONTERO a María-Cristina RIFF. On-the-fly calibrating strategies for evolutionary algorithms, *Information Sciences*, Volume 181, Issue 3/1, ISSN 0020-0255, 10.1016/j.ins.2010.09.016, (2011), 552-566.
- [169] A. E. EIBEN, E. MARCHIORI, and V. A. VALKÓ. Evolutionary algorithms with on-the-fly population size adjustment. In *Parallel Problem Solving from Nature, PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*. Springer, (2004), 41–50
- [170] BREST J a MAUCEC MS. Population size reduction for the differential evolution algorithm. *Applied Intelligence*, issue. 29, n. 3. ISSN 0924-669x, (2008)
- [171] VINOD V., SRIDHARAN R. Scheduling a dynamic job shop production system with sequence-dependent setups: An experimental study *Source, Robotics and Computer-Integrated Manufacturing archive*, Volume 24 , Issue 3 (June 2008) Pages 435-449 ISSN:0736-5845

Přehled publikační činnosti

Články v recenzovaných časopisech:

- KOBLASA, F., MANLIG, F., VAVRUŠKA, J. Evolution Algorithm for Job Shop Scheduling Problem Constrained by the Optimization Timespan. Applied Mechanics and Materials. 2013, vol. 309, s. 350-357. DOI: 10.4028/www.scientific.net/AMM.309.350. Dostupné z: <http://www.scientific.net/AMM.309.350>
- KLOUD, T; KOBLASA, F. Solving Job Shop scheduling with the computer simulation . Doprava a Logistika (Transport and Logistics). 27-30.10.2011, 9th Special Issue, s. 775-785. ISSN 1451-107X.
- PLECHAČ, R. - MANLIG, F. – KOBLASA, F. Job Shop Scheduling With Aid of Computer Simulation. University Review Vol. 4, No. 2. Trenčín: Alexander Dubček University of Trenčín, 2010, s. 29-33 ISSN 1337-6047
- KOBLASA F.,MANLIG F. Job shop scheduling problem with multi-shift work system. In:“Vědecká pojednání” XV / 2009, ACC Journal, č. publikace: 55-050-09, čj. 50/09, Technická univerzita v Liberci, ISSN 1803-9782,
- KOBLASA F., DIAS L.S., OLIVEIRA J.A. Scheduling Optimization Using Local Search and Genetic Algorithm”. “Vědecká pojednání” XIV / 2008, ACC Journal, ISBN 978-80-7372-379-8, ISSN 1801-1128, č. publikace: 55-083-08, čj. RE 106/08, Technická univerzita v Liberci.
- MANLIG, F.; VAVRUŠKA, J.; KOBLASA, F. Möglichkeiten der Zusammenarbeit auf dem Gebiet der Produktionssysteme. In. Vědecká pojednání/Wissenschaftliche Abhandlungen/Práce naukowe - Akademické koordinační středisko v Euroregionu Nisa. TU v Liberci, duben 2007, s. 89-92 ISSN 978-80-7372-195-4
- VAVRUŠKA J., KOBLASA F. The constrained production capacities and its planning. ACC Journal. Vědecká pojednání/Wissenschaftliche Abhandlungen/Práce naukowe - Akademické koordinační středisko v Euroregionu Nisa. r. XIII, 2007. ISSN 1801-1128
- GOTTWALDOVÁ A., KOBLASA F., Die Verwendug der Simulationsspiele im Unterricht. ACC Journal. Vědecká pojednání/Wissenschaftliche Abhandlungen/Práce naukowe - Akademické koordinační středisko v Euroregionu Nisa. r. XII, 2006. ISSN 1801-1128
- VAVRUŠKA J., KOBLASA F. THE USE OF SIMULATION IN THE WORKSHOP PLANNING AREA. ACC Journal. Vědecká pojednání/Wissenschaftliche Abhandlungen/Práce naukowe - Akademické koordinační středisko v Euroregionu Nisa. r. XII, 2006. ISSN 1801-1128
- GOTTWALDOVÁ A., KOBLASA F., Die Verwendug der Simulationsspiele im Unterricht. ACC Journal. Vědecká pojednání/Wissenschaftliche Abhandlungen/Práce naukowe - Akademické koordinační středisko v Euroregionu Nisa. r. XII, 2006. ISSN 1801-1128

Mezinárodní konference

- KOBLASA F. Flexible job shop scheduling and optimizing by genetic algorithm in the Simcron MODELLER. In: Sborník příspěvků 13. ročník mezinárodní konference WITNESS 2010, 3.6-4.6 Kouty nad Desnou, ISBN 978-80-214-4107-1
- VAVRUŠKA J., MANLIG F., KOBLASA F., ORDORICA M.: Analýza rozvrhování pracovníků a výrobní mix. In: Sborník příspěvků 12. ročník mezinárodní konference WITNESS 2009, ISBN 978-80-214-3900-9
- VAVRUŠKA J., MANLIG F., KOBLASA F.: One piece flow – Caravans, deeper recognition. In: International doctoral seminar, Smolenice květen 2009, ISBN 978-80-8096-088-9
- KOBLASA F., VAVRUŠKA J.: Job shop scheduling and heuristic optimization. In: International doctoral seminar, Smolenice květen 2009, ISBN 978-80-8096-088-9
- KOBLASA F., DIAS L.S., OLIVEIRA J.A., PEREIRA G. Heuristic Approach as a way to Improve Scheduling in ERP/APS Systems". Proceedings of 15th European Concurrent Engineering Conference (ECEC2008). Eds. A. Brito and J.M. Teixeira, 47-51, Porto April 2008. EUROSIS-ETI Publication. ISBN 978-9077381-399-7. (All EUROSIS Proceedings are ISI-Thomson and INSPEC referenced)
- KOBLASA F., DIAS L.S., OLIVEIRA J.A. Simulation and possibility of optimization in ERP/APS systems". Proceedings of 4th annual International conference for Ph.D students and young researches, Zlín 10. April 2008, ISBN 978-80-7318-663-0
- VAVRUŠKA J., KOBLASA F. Planning concepts and their use in the planning applications. 5. ročník mezinárodní konference Finanční a logistické řízení 2007. VŠB - Technická univerzita Ostrava 2007. ISBN 978-80-248-1406-3
- VAVRUŠKA, J.; KOBLASA, F. Variety of operation planning and their effect on planning applications. V. International Scientific Conference Management, Economics and Business development in the new European conditions. VUT v Brně 2007, s.103. ISBN 978-80-7204-532-7
- KOBLASA, F.; VAVRUŠKA, J.; VIK, P.: Innovation of the information system at KVS In: Setkání ústavů a kateder výrobních strojů a robotizace. Sborník příspěvků mezinárodní konference, Západočeská univerzita v Plzni. Plzeň 10.-11. 9. 2007, ISBN 978-80-7043-598-4
- KOBLASA F., DIAS L.S., OLIVEIRA J.A. Using heuristic to improve scheduling in the ERP/APS systems. Proceedings of 2nd international conference "Výrobní systémy dnes a zítra 2007". Liberec 6-7 December 2007. ISBN 978-80-7372-295-1.
- MANLIG, F.; VAVRUŠKA, J.; KOBLASA, F. Möglichkeiten der Zusammenarbeit auf dem Gebiet der Produktionssysteme. Přednáška na mezinárodní konferenci Międzynarodowa Konferencja ACC Rola nauki w rozwoju Euroregionu Nysa – stan i perspektywy. Karpacz 19 – 20 października 2006

Kapitola v knize

- MANLIG, F; LADA, O ; KOBLASA, F. DAAM International scientific book 2011. vol. 10. Vienna (Austria) : [s.n.], 2011. THE EXPERIENCES WITH REENGINEERING USING COMPUTER SIMULATION, s. v tisku. ISBN 978-3901509-84-1, ISSN: 1726-9687 cross ref DOI: 10.2507.7

Ostatní konference

- KOBLASA, F. Uplatnění heuristických optimalizačních metod v oblasti rozvrhování strojírenské výroby SME podniků. Workshop pro doktorandy Fakulty Strojní a Fakulty Textilní Technické univerzity v Liberci 17-20 září: Sekce STROJÍRENSTVÍ. ISBN 978-80-7372-891-5.
- KOBLASA F. Single swap local search for classical jobshop and flexible job shop scheduling problem. In: Recenzovaný sborník abstraktů z Mezinárodní Baťovy konference pro doktorandy a mladé vědecké pracovníky 2010. Univerzita Tomáše Bati ve Zlíně. 15.4.2010. ISBN 978-80-7318-922-8.
- KOBLASA F. Constrained scheduling - shifts and setups. In: Recenzovaný sborník abstraktů z Mezinárodní Baťovy konference pro doktorandy a mladé vědecké pracovníky 2009. Univerzita Tomáše Bati ve Zlíně. 2009. ISBN 978-80-7318-811-5
- PLECHAČ R., KOBLASA F. FLEXIBLE JOB SHOP SCHEDULING USING SIMULATION SOFTWARE SIMCRON MODELLER. In: 4. ročník mezinárodní konference „Výrobní systémy dnes a zítra 2009“. Sborník anotací příspěvků. TU v Liberci 19. - 20. 11. 2009.
- KOBLASA F. GIFFLER AND THOMPSON ALGORITHM FOR FLEXIBLE JOB SHOP PROBLEM. In: 4. ročník mezinárodní konference „Výrobní systémy dnes a zítra 2009“. Sborník anotací příspěvků. TU v Liberci 19. - 20. 11. 2009."
- KOBLASA, F. Uplatnění optimalizačních metod v dílenském rozvrhování výrobních zakázek. In: Výrobní systémy dnes a zítra 2008. Sborník anotací příspěvků z mezinárodní konference, Liberec 27. - 28. 11. 2008. Liberec: TU v Liberci - KVS, 2008. ISBN 978-80-7372-416-0
- KOBLASA, F. Modified Giffler and Thompson algorithm for setup and pass-setup constrain. In: 3. ročník mezinárodní konference „Výrobní systémy dnes a zítra 2008“. Sborník anotací příspěvků, Liberec 27. - 28. 12. 2008. Liberec: TU v Liberci - KVS, 2008. ISBN 978-80-7372-416-0.
- VAVRUŠKA J., KOBLASA F. The contributions of APS and simulation in the production planning. Mezinárodní Baťovy doktorské konference Recenzovaný sborník abstraktů z konference studentů doktorského studijního programu. Univerzita Tomáše Bati ve Zlíně. 2007. ISBN 978-80-7318-529-9

Publikovaný software

- KOBLASA F. Nástroj pro rozvrhování zakázkové výroby, Liberec: Technická univerzita v Liberci, 2012

Seznam příloh

Příloha č. 1: Plnění zákaznických termínů 2/3 modelu TOS VARNSDORF a.s.

Příloha č. 2: Rozdíl mezi zákaznickým termínem a rozvrženým termínem plného modelu TOS VARNSDORF a.s.

Příloha č. 3: Využití pracovišť vybraného řešení plného modelu TOS VARNSDORF a.s.

Příloha č. 4: Plnění zákaznických termínů 2/3 modelu MZ Liberec a.s.

Příloha č. 5a,b: Plnění zákaznických termínů plného modelu MZ Liberec a.s.

Příloha č. 6a,b: Využití pracovišť vybraného řešení plného modelu MZ Liberec a.s.

Příloha č. 7: Rozvrhovací nástroj

Příloha č. 1: Plnění zákaznických termínů 2/3 modelu TOS VARNSDORF a.s.

Zakázka	Termín náběhu	Zákaznický termín	Rozvrhovaný termín	Rozdíl [dd:hh:mm:ss]
0	16.12.2011 6:00:00	5.1.2012 22:00	28.12.2011 22:48	- 07.23:12:00
1	25.11.2011 6:00:00	5.1.2012 22:00	14.12.2011 20:30	- 22.01:30:00
2	25.10.2011 6:00:00	9.1.2012 22:00	9.12.2011 1:36	- 31.20:24:00
3	9.12.2011 6:00:00	5.1.2012 22:00	25.12.2011 22:48	- 10.23:12:00
4	21.11.2011 6:00:00	5.1.2012 22:00	7.12.2011 13:46	- 29.08:14:00
5	21.11.2011 6:00:00	5.1.2012 22:00	18.12.2011 17:30	- 18.04:30:00
6	2.12.2011 6:00:00	5.1.2012 22:00	21.12.2011 0:28	- 15.21:32:00
7	14.10.2011 6:00:00	30.10.2011 22:00	19.10.2011 3:22	- 11.18:38:00
8	9.12.2011 6:00:00	5.1.2012 22:00	19.12.2011 5:59	- 17.16:01:00
9	26.12.2011 6:00:00	9.1.2012 22:00	5.1.2012 1:00	- 04.21:00:00
10	16.12.2011 6:00:00	5.1.2012 22:00	28.12.2011 2:55	- 08.19:05:00
11	14.12.2011 6:00:00	9.1.2012 22:00	25.12.2011 18:03	- 15.03:57:00
12	10.10.2011 6:00:00	30.10.2011 22:00	21.10.2011 5:00	- 09.17:00:00
13	9.12.2011 6:00:00	5.1.2012 22:00	19.12.2011 0:37	- 17.21:23:00
14	2.12.2011 6:00	9.1.2012 22:00	14.1.2012 14:01	04.16:01:00
15	2.12.2011 6:00	9.1.2012 22:00	22.1.2012 2:06	12.04:06:00
16	8.12.2011 6:00	9.1.2012 22:00	20.1.2012 23:36	11.01:36:00
17	6.12.2011 6:00	9.1.2012 22:00	5.1.2012 1:45	- 04.20:15:00
18	11.12.2011 6:00	9.1.2012 22:00	1.1.2012 23:58	- 07.22:02:00
19	15.12.2011 6:00	27.2.2012 22:00	27.1.2012 1:36	- 31.20:24:00
20	9.12.2011 6:00	9.1.2012 22:00	24.12.2011 4:13	- 16.17:47:00
21	23.11.2011 6:00	9.1.2012 22:00	13.12.2011 10:34	- 27.11:26:00
22	5.12.2011 6:00	9.1.2012 22:00	26.12.2011 21:23	- 14.00:37:00
23	14.2.2012 6:00	21.2.2012 22:00	27.2.2012 22:48	06.00:48:00
24	28.12.2011 6:00	9.1.2012 22:00	8.1.2012 0:49	- 01.21:11:00
25	8.2.2012 6:00	27.2.2012 22:00	19.2.2012 18:33	- 08.03:27:00
26	8.12.2011 6:00	9.1.2012 22:00	3.1.2012 22:30	- 05.23:30:00
27	14.12.2011 6:00	9.1.2012 22:00	22.12.2011 20:05	- 18.01:55:00
28	14.12.2011 6:00	9.1.2012 22:00	23.12.2011 2:30	- 17.19:30:00
29	7.12.2011 6:00	9.1.2012 22:00	24.12.2011 22:42	- 15.23:18:00
30	15.11.2011 6:00	5.1.2012 22:00	4.12.2011 1:36	- 32.20:24:00
31	30.11.2011 6:00:00	5.1.2012 22:00	7.1.2012 5:06	01.07:06:00
32	16.11.2011 6:00:00	5.1.2012 22:00	4.12.2011 20:36	- 32.01:24:00

Příloha č. 2: Rozdíl mezi zákaznickým termínem a rozvrženým termínem plného modelu TOS VARNSDORF a.s.

Číslo zakázky	Termín stanovený zákazníkem	Rozdíl mezi rozvrhovaným termínem a termínem zákazníka [dny.hodiny:minuty:sekundy]	Číslo zakázky	Termín stanovený zákazníkem	Rozdíl mezi rozvrhovaným termínem a termínem zákazníka [dny.hodiny:minuty:sekundy]
0	5.12.2011	- 16.23:12:00	42	5.12.2011	- 28.10:42:00
1	5.12.2011	03.23:45:00	43	5.12.2011	- 20.11:00:00
2	5.12.2011	20.03:36:00	44	5.12.2011	- 22.20:40:00
3	5.12.2011	- 12.22:44:00	45	5.12.2011	05.21:30:00
4	5.12.2011	- 09.08:14:00	46	5.12.2011	- 06.22:47:00
5	5.12.2011	01.19:30:00	47	5.12.2011	- 11.15:25:00
6	5.12.2011	- 09.03:41:00	48	5.12.2011	13.12:47:00
7	5.12.2011	- 24.21:46:00	49	5.12.2011	- 21.20:22:00
8	5.12.2011	- 17.17:12:00	50	5.12.2011	- 29.21:22:00
9	5.12.2011	- 20.21:00:00	51	5.12.2011	- 29.17:54:00
10	5.12.2011	- 18.19:21:00	52	5.12.2011	- 29.02:57:00
11	5.12.2011	- 14.10:46:00	53	5.12.2011	- 10.21:46:00
12	5.12.2011	- 18.16:49:00	54	5.12.2011	- 28.03:04:00
13	5.12.2011	- 20.20:32:00	55	5.12.2011	- 23.09:50:00
14	5.12.2011	09.12:53:00	56	5.12.2011	- 30.00:01:00
15	5.12.2011	18.04:06:00	57	5.12.2011	- 29.10:13:00
16	5.12.2011	14.01:09:00	58	5.12.2011	- 19.21:26:00
17	5.12.2011	17.00:54:00	59	5.12.2011	- 21.22:17:00
18	5.12.2011	- 03.17:05:00	60	5.12.2011	- 27.22:45:00
19	5.12.2011	20.08:12:00	61	5.12.2011	- 22.00:43:00
20	5.12.2011	- 12.01:17:00	62	5.12.2011	- 28.01:06:00
21	5.12.2011	- 07.05:33:00	63	5.12.2011	- 18.02:49:00
22	5.12.2011	- 06.21:02:00	64	5.12.2011	- 20.02:18:00
23	5.12.2011	- 14.18:30:00	65	5.12.2011	- 27.23:18:00
24	5.12.2011	- 11.22:48:00	66	5.12.2011	- 29.19:02:00
25	5.12.2011	- 13.08:54:00	67	5.12.2011	- 21.12:27:00
26	5.12.2011	00.00:30:00	68	5.12.2011	- 17.03:30:00
27	5.12.2011	- 14.22:30:00	69	5.12.2011	- 28.21:24:00
28	5.12.2011	- 22.21:10:00	70	5.12.2011	- 17.13:35:00
29	5.12.2011	07.03:37:00	71	5.12.2011	- 22.23:41:00
30	5.12.2011	- 00.16:24:00	72	5.12.2011	- 19.22:27:00
31	5.12.2011	01.07:01:00	73	5.12.2011	- 26.00:15:00
32	5.12.2011	- 07.18:37:00	74	5.12.2011	- 14.18:38:00
33	5.12.2011	- 27.21:53:00	75	5.12.2011	- 14.20:06:00
34	5.12.2011	- 04.20:35:00	76	5.12.2011	- 21.21:21:00
35	5.12.2011	- 10.21:07:00	77	5.12.2011	- 29.20:20:00
36	5.12.2011	- 28.22:25:00	78	5.12.2011	- 11.01:31:00
37	5.12.2011	- 28.22:44:00	79	5.12.2011	- 16.03:10:00
38	5.12.2011	- 16.21:43:00	80	5.12.2011	- 14.01:22:00
39	5.12.2011	- 19.01:20:00	81	5.12.2011	- 02.07:40:00
40	5.12.2011	- 06.14:50:00	82	5.12.2011	- 26.09:56:00
41	5.12.2011	06.05:20:00	83	5.12.2011	- 25.10:48:00

**Příloha č. 3: Využití pracovišť vybraného řešení plného modelu TOS
VARNSDORF a.s.**

Stroj	Stroj je ve směně	Stroj je mimo směnu	Celkem - pracuje	Ve směně - pracuje	Celkem - seřizování	Ve směně - seřizování	Celkem - nepracuje	Ve směně - nepracuje	Ve směně - práce je blokována	Mimo směnu - práce je blokována	Ve směně - seřizování je blokováno	Mimo směnu - seřizování je blokováno
0	63%	37%	1%	1%	0%	0%	62%	99%	0%	0%	0%	0%
1	31%	69%	18%	57%	2%	8%	11%	35%	32%	46%	20%	29%
2	31%	69%	3%	8%	1%	2%	28%	90%	1%	2%	12%	17%
3	31%	69%	2%	7%	1%	2%	28%	91%	3%	4%	1%	2%
4	31%	69%	13%	43%	3%	10%	15%	47%	17%	25%	25%	37%
5	63%	37%	23%	37%	3%	4%	37%	58%	11%	31%	6%	16%
6	63%	37%	28%	44%	3%	5%	31%	50%	18%	49%	1%	4%
7	94%	6%	66%	71%	4%	4%	24%	25%	4%	71%	0%	4%
8	31%	69%	0%	0%	0%	0%	31%	100%	0%	0%	0%	0%
9	31%	69%	2%	6%	0%	1%	29%	92%	1%	2%	7%	10%
10	31%	69%	0%	1%	0%	1%	31%	99%	0%	0%	4%	6%
11	31%	69%	5%	16%	1%	2%	26%	82%	7%	10%	13%	19%
12	31%	69%	4%	14%	1%	4%	26%	83%	3%	4%	16%	23%
13	31%	69%	13%	42%	2%	7%	16%	51%	21%	31%	24%	35%
14	31%	69%	15%	49%	3%	8%	13%	43%	20%	29%	32%	46%
15	31%	69%	1%	4%	0%	1%	30%	95%	0%	0%	8%	12%
16	31%	69%	4%	14%	1%	4%	26%	83%	0%	0%	28%	40%
17	31%	69%	1%	4%	1%	2%	29%	94%	0%	0%	15%	21%
18	31%	69%	2%	5%	0%	1%	29%	94%	1%	2%	4%	6%
19	31%	69%	24%	76%	2%	6%	6%	18%	38%	56%	22%	33%
20	31%	69%	0%	0%	0%	0%	31%	100%	0%	0%	0%	0%
21	100%	0%	1%	1%	0%	0%	99%	99%	0%	0%	0%	0%
22	100%	0%	1%	1%	0%	0%	99%	99%	0%	0%	0%	0%
23	100%	0%	1%	1%	0%	0%	99%	99%	0%	0%	0%	0%
24	100%	0%	1%	1%	0%	0%	99%	99%	0%	0%	0%	0%
25	100%	0%	3%	3%	0%	0%	97%	97%	0%	0%	0%	0%
26	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
27	100%	0%	0%	0%	0%	0%	99%	99%	0%	0%	0%	0%
28	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
29	100%	0%	1%	1%	0%	0%	99%	99%	0%	0%	0%	0%
30	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
31	100%	0%	3%	3%	0%	0%	97%	97%	0%	0%	0%	0%
32	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
33	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
34	100%	0%	7%	7%	0%	0%	93%	93%	0%	0%	0%	0%
35	100%	0%	8%	8%	0%	0%	92%	92%	0%	0%	0%	0%
36	100%	0%	1%	1%	0%	0%	99%	99%	0%	0%	0%	0%

Příloha č. 4: Plnění zákaznických termínů 2/3 modelu MZ Liberec a.s.

Číslo zakázky	Termín stanovený zákazníkem	Termín reálného odvedení	Rozdíl mezi rozvrhovaným termínem a termínem zákazníka [dny:hodiny:minuty:sekundy]	Číslo zakázky	Termín stanovený zákazníkem	Termín reálného odvedení	Rozdíl mezi rozvrhovaným termínem a termínem zákazníka [dny:hodiny:minuty:sekundy]
1	31.1.2013 6:00	21.5.2013 11:10	110.05:10:00	77	10.2.2013 6:00	8.2.2013 22:46	- 01.07:14:00
2	31.1.2013 6:00	23.1.2013 11:08	- 07.18:52:00	78	4.2.2013 6:00	22.5.2013 18:41	107.12:41:00
3	31.1.2013 6:00	23.1.2013 11:37	- 07.18:23:00	79	31.1.2013 6:00	13.5.2013 21:05	102.15:05:00
4	31.1.2013 6:00	23.1.2013 12:06	- 07.17:54:00	80	31.1.2013 6:00	14.5.2013 8:54	103.02:54:00
5	15.2.2013 6:00	25.1.2013 11:01	- 20.18:59:00	81	31.1.2013 6:00	23.1.2013 18:51	- 07.11:09:00
6	13.2.2013 6:00	1.2.2013 15:23	- 11.14:37:00	82	31.1.2013 6:00	24.1.2013 14:01	- 06.15:59:00
7	1.2.2013 6:00	4.2.2013 5:32	02.23:32:00	83	13.2.2013 6:00	1.2.2013 7:54	- 11.22:06:00
8	1.2.2013 6:00	8.2.2013 12:13	07.06:13:00	84	4.3.2013 6:00	6.6.2013 4:03	93.22:03:00
9	31.1.2013 6:00	8.2.2013 7:25	08.01:25:00	85	31.1.2013 6:00	4.6.2013 0:02	123.18:02:30
10	7.2.2013 6:00	29.1.2013 16:41	- 08.13:18:48	86	1.2.2013 6:00	6.2.2013 22:01	05.16:01:00
11	31.1.2013 6:00	31.1.2013 14:30	00.08:30:00	87	30.1.2013 6:00	26.1.2013 13:17	- 03.16:43:00
12	31.1.2013 6:00	1.2.2013 6:30	01.00:30:00	88	2.2.2013 6:00	16.5.2013 11:51	103.05:51:54
13	7.2.2013 6:00	2.2.2013 6:10	- 04.23:49:48	89	1.2.2013 6:00	31.1.2013 12:38	- 00.17:22:00
14	31.1.2013 6:00	24.1.2013 18:45	- 06.11:15:00	90	10.2.2013 6:00	3.2.2013 6:06	- 06.23:54:00
15	31.1.2013 6:00	23.5.2013 17:15	112.11:15:00	91	41315,25	41308,25049	- 06.23:59:18
16	30.1.2013 6:00	1.6.2013 17:30	122.11:30:00	92	41316,25	41302,99514	- 13.06:07:00
17	3.2.2013 6:00	26.1.2013 18:54	- 07.11:06:00	93	41312,25	41315,72	03.11:16:48
18	31.1.2013 6:00	25.1.2013 6:56	- 05.23:03:42	94	41312,25	41315,80347	03.13:17:00
19	1.2.2013 6:00	31.1.2013 10:11	- 00.19:48:24	95	41314,25	41312,42816	- 01.19:43:26
20	7.2.2013 6:00	26.1.2013 10:05	- 11.19:54:12	96	41314,25	41312,51635	- 01.17:36:26
21	31.1.2013 6:00	26.1.2013 15:33	- 04.14:27:00	97	41314,25	41312,54177	- 01.16:59:50
22	1.2.2013 6:00	31.1.2013 10:44	- 00.19:15:12	98	41319,25	41422,7324	103.11:34:39
23	7.2.2013 6:00	26.1.2013 15:49	- 11.14:10:06	99	41305,25	41403,76181	98.12:17:00
24	12.2.2013 6:00	29.5.2013 11:20	106.05:20:00	100	41313,25	41305,44722	- 07.19:16:00
25	12.2.2013 6:00	31.5.2013 9:46	108.03:46:00	101	41314,25	41304,5875	- 09.15:54:00
26	1.2.2013 6:00	31.1.2013 10:27	- 00.19:32:36	102	41306,25	41310,73604	04.11:39:54
27	31.1.2013 6:00	15.2.2013 17:43	15.11:43:00	103	41300,25	41299,26111	- 00.23:44:00
28	31.1.2013 6:00	12.2.2013 16:03	12.10:03:00	104	41315,25	41300,26806	- 14.23:34:00
29	4.3.2013 6:00	28.1.2013 11:32	- 34.18:28:00	105	41301,25	41300,27569	- 00.23:23:00
30	24.1.2013 6:00	23.1.2013 6:03	- 00.23:57:00	106	41312,25	41300,39583	- 11.20:30:00
31	24.1.2013 6:00	23.1.2013 12:19	- 00.17:41:00	107	41319,25	41306,37708	- 12.20:57:00
32	4.3.2013 6:00	26.1.2013 6:03	- 36.23:57:00	108	41315,25	41302,61042	- 12.15:21:00
33	24.1.2013 6:00	23.1.2013 12:32	- 00.17:28:00	109	41318,25	41313,51042	- 04.17:45:00
34	24.1.2013 6:00	23.1.2013 12:45	- 00.17:15:00	110	41305,25	41303,65278	- 01.14:20:00
35	11.3.2013 6:00	19.1.2013 6:13	- 50.23:47:00	111	41305,25	41303,79167	- 01.11:00:00
36	5.2.2013 6:00	4.5.2013 19:40	88.13:40:00	112	41305,25	41303,93056	- 01.07:40:00
37	31.1.2013 6:00	30.1.2013 4:10	- 01.01:50:00	113	41305,25	41304,06944	- 01.04:20:00
38	1.2.2013 6:00	7.2.2013 6:42	06.00:42:00	114	41305,25	41306,66319	01.09:55:00
39	8.2.2013 6:00	23.1.2013 6:10	- 15.23:50:00	115	41305,25	41306,80208	01.13:15:00
40	10.2.2013 6:00	24.1.2013 19:25	- 16.10:35:00	116	41305,25	41306,94097	01.16:35:00
41	20.2.2013 6:00	26.5.2013 16:01	95.10:01:00	117	41305,25	41307,07986	01.19:55:00
42	1.2.2013 6:00	9.6.2013 18:51	128.12:51:00	118	41305,25	41304,32292	- 00.22:15:00
43	31.1.2013 6:00	18.5.2013 9:10	107.03:10:00	119	41305,25	41301,04792	- 04.04:51:00
44	12.2.2013 6:00	2.6.2013 22:58	110.16:58:00	120	41305,25	41299,725	- 05.12:36:00
45	31.1.2013 6:00	17.5.2013 23:04	106.17:04:00	121	41296,25	41297,725	01.11:24:00
46	31.1.2013 6:00	29.1.2013 2:40	- 02.03:20:00	122	41305,25	41302,47917	- 02.18:30:00
47	28.1.2013 6:00	25.1.2013 16:00	- 02.14:00:00	123	41300,25	41295,68125	- 04.13:39:00
48	8.2.2013 6:00	31.1.2013 6:01	- 07.23:59:00	124	41305,25	41311,72778	06.11:28:00
49	31.1.2013 6:00	23.1.2013 7:40	- 07.22:20:00	125	41320,25	41301,6875	- 18.13:30:00
50	1.2.2013 6:00	4.6.2013 12:51	123.06:51:00	126	41320,25	41302,46528	- 17.18:50:00
51	31.1.2013 6:00	17.2.2013 17:43	17.11:43:00	127	41312,25	41300,43611	- 11.19:32:00
52	31.1.2013 6:00	29.1.2013 5:30	- 02.00:30:00	128	41322,25	41304,68958	- 17.13:27:00
53	10.2.2013 6:00	14.5.2013 9:28	93.03:28:00	129	41313,25	41305,43819	- 07.19:29:00
54	4.3.2013 6:00	17.5.2013 6:30	74.00:30:00	130	41314,25	41298,86042	- 15.09:21:00
55	1.2.2013 6:00	7.2.2013 16:03	06.10:03:48	131	41305,25	41302,55139	- 02.16:46:00
56	31.1.2013 6:00	25.1.2013 14:50	- 05.15:10:00	132	41305,25	41298,71875	- 06.12:45:00
57	31.1.2013 6:00	27.1.2013 7:04	- 03.22:56:00	133	41313,25	41300,26806	- 12.23:34:00
58	1.2.2013 6:00	6.2.2013 18:18	05.12:18:00	134	41311,25	41298,04861	- 13.04:50:00
59	31.1.2013 6:00	28.1.2013 2:10	- 03.03:50:00	135	41311,25	41298,22222	- 13.00:40:00
60	7.2.2013 6:00	24.5.2013 16:11	106.10:11:00	136	41312,25	41307,41833	- 04.19:57:36
61	7.2.2013 6:00	26.5.2013 16:26	108.10:26:00	137	41305,25	41314,52847	09.06:41:00
62	20.2.2013 6:00	29.1.2013 2:06	- 22.03:54:00	138	41305,25	41318,59722	13.08:20:00
63	4.3.2013 6:00	3.2.2013 6:15	- 28.23:44:24	139	41314,25	41301,42813	- 12.19:43:30
64	31.1.2013 6:00	3.2.2013 17:30	03.11:30:00	140	41316,25	41310,90417	- 05.08:18:00
65	9.2.2013 6:00	23.1.2013 9:09	- 16.20:50:24	141	41312,25	41300,26417	- 11.23:39:36
66	13.2.2013 6:00	24.1.2013 14:02	- 19.15:57:30	142	41312,25	41300,25556	- 11.23:52:00
67	23.1.2013 6:00	20.1.2013 10:40	- 02.19:20:00	143	41306,25	41305,80694	- 00.10:38:00
68	10.2.2013 6:00	23.1.2013 8:11	- 17.21:49:00	144	41312,25	41300,25292	- 11.23:55:48
69	1.2.2013 6:00	3.2.2013 10:00	02.04:00:18	145	41305,25	41414,77806	109.12:40:24
70	31.1.2013 6:00	2.2.2013 12:30	02.06:30:00	146	41318,25	41424,86458	106.14:45:00
71	31.1.2013 6:00	30.1.2013 9:10	- 00.20:50:00	147	41312,25	41307,25139	- 04.23:58:00
72	31.1.2013 6:00	18.5.2013 22:52	107.16:52:48	148	41305,25	41315,20833	09.23:00:00
73	1.2.2013 6:00	7.2.2013 12:36	06.06:36:00	149	41315,25	41300,26111	- 14.23:44:00
74	1.2.2013 6:00	4.6.2013 14:26	123.08:26:00	150	41315,25	41300,27014	- 14.23:31:00
75	1.2.2013 6:00	4.2.2013 12:53	03.06:53:48	151	41302,25	41299,25208	- 02.23:57:00
76	1.2.2013 6:00	1.2.2013 5:59	- 00.00:01:00	152	41314,25	41297,70347	- 16.13:07:00

Příloha č. 5a: Plnění zákaznických termínů plného modelu MZ Liberec a.s.

Číslo zakázky	Termín stanovený zákazníkem	Termín reálného odvedení	Rozdíl mezi rozvrhovaným termínem a termínem zákazníka [dny:hodiny:minuty:sekundy]	Číslo zakázky	Termín stanovený zákazníkem	Termín reálného odvedení	Rozdíl mezi rozvrhovaným termínem a termínem zákazníka [dny:hodiny:minuty:sekundy]
1	31.1.2013 6:00	31.1.2013 5:38	- 00.00:21:30	59	27.1.2013 6:00	27.1.2013 3:05	- 00.02:55:00
2	26.1.2013 6:00	23.1.2013 11:00	- 02.19:00:00	60	3.2.2013 6:00	30.1.2013 3:25	- 04.02:35:00
3	26.1.2013 6:00	23.1.2013 11:29	- 02.18:31:00	61	3.2.2013 6:00	31.1.2013 21:32	- 02.08:28:00
4	26.1.2013 6:00	23.1.2013 13:16	- 02.16:44:00	62	29.1.2013 6:00	28.1.2013 21:38	- 00.08:22:00
5	1.2.2013 6:00	25.1.2013 11:01	- 06.18:59:00	63	4.2.2013 6:00	3.2.2013 11:28	- 00.18:32:00
6	3.2.2013 6:00	1.2.2013 10:50	- 01.19:10:00	64	2.2.2013 6:00	4.2.2013 17:36	02.11:36:00
7	1.2.2013 6:00	1.2.2013 3:51	- 00.02:09:00	65	24.1.2013 6:00	23.1.2013 9:09	- 00.20:50:24
8	1.2.2013 6:00	26.1.2013 4:17	- 06.01:43:00	66	25.1.2013 6:00	24.1.2013 14:02	- 00.15:57:30
9	8.2.2013 6:00	8.2.2013 7:24	00.01:24:42	67	21.1.2013 6:00	20.1.2013 10:40	- 00.19:20:00
10	27.1.2013 6:00	26.1.2013 18:34	- 00.11:25:48	68	24.1.2013 6:00	23.1.2013 8:11	- 00.21:49:00
11	31.1.2013 6:00	31.1.2013 18:12	00.12:12:00	69	1.2.2013 6:00	3.2.2013 7:51	02.01:51:00
12	31.1.2013 6:00	30.1.2013 15:25	- 00.14:35:00	70	31.1.2013 6:00	9.2.2013 12:56	09.06:56:00
13	3.2.2013 6:00	2.2.2013 6:10	- 00.23:49:48	71	31.1.2013 6:00	1.2.2013 3:41	00.21:41:00
14	25.1.2013 6:00	24.1.2013 19:25	- 00.10:35:00	72	29.1.2013 6:00	30.1.2013 14:53	01.08:53:30
15	31.1.2013 6:00	1.2.2013 8:26	01.02:26:48	73	4.2.2013 6:00	7.2.2013 13:58	03.07:58:30
16	14.2.2013 6:00	10.2.2013 21:26	- 03.08:34:00	74	4.2.2013 6:00	3.2.2013 22:31	- 00.07:29:00
17	29.1.2013 6:00	26.1.2013 18:54	- 02.11:06:00	75	4.2.2013 6:00	5.2.2013 4:17	00.22:17:00
18	26.1.2013 6:00	25.1.2013 6:56	- 00.23:03:42	76	1.2.2013 6:00	31.1.2013 8:07	- 00.21:52:30
19	1.2.2013 6:00	31.1.2013 10:24	- 00.19:35:54	77	9.2.2013 6:00	9.2.2013 19:00	00.13:00:00
20	27.1.2013 6:00	26.1.2013 13:12	- 00.16:47:12	78	31.1.2013 6:00	29.1.2013 6:11	- 01.23:48:30
21	27.1.2013 6:00	26.1.2013 16:08	- 00.13:51:06	79	24.1.2013 6:00	24.1.2013 18:01	00.12:01:30
22	1.2.2013 6:00	31.1.2013 10:05	- 00.19:54:12	80	25.1.2013 6:00	24.1.2013 17:08	- 00.12:51:18
23	27.1.2013 6:00	26.1.2013 14:03	- 00.15:56:06	81	24.1.2013 6:00	23.1.2013 21:01	- 00.08:59:00
24	12.2.2013 6:00	16.2.2013 4:13	03.22:13:00	82	25.1.2013 6:00	24.1.2013 16:11	- 00.13:49:00
25	12.2.2013 6:00	12.2.2013 5:41	- 00.00:19:00	83	13.2.2013 6:00	4.2.2013 5:42	- 09.00:18:00
26	1.2.2013 6:00	31.1.2013 16:11	- 00.13:48:12	84	14.2.2013 6:00	8.2.2013 23:01	- 05.06:59:00
27	13.2.2013 6:00	13.2.2013 11:37	00.05:37:00	85	31.1.2013 6:00	3.2.2013 21:13	03.15:13:00
28	13.2.2013 6:00	14.2.2013 6:39	01.00:39:48	86	5.2.2013 6:00	1.2.2013 5:06	- 04.00:54:00
29	29.1.2013 6:00	28.1.2013 11:32	- 00.18:28:00	87	25.1.2013 6:00	24.1.2013 23:45	- 00.06:15:00
30	24.1.2013 6:00	23.1.2013 12:34	- 00.17:26:00	88	28.1.2013 6:00	26.1.2013 8:26	- 01.21:34:00
31	24.1.2013 6:00	23.1.2013 11:42	- 00.18:18:00	89	1.2.2013 6:00	31.1.2013 17:17	- 00.12:42:12
32	27.1.2013 6:00	26.1.2013 6:03	- 00.23:57:00	90	3.2.2013 6:00	2.2.2013 14:53	- 00.15:06:42
33	24.1.2013 6:00	23.1.2013 11:55	- 00.18:05:00	91	3.2.2013 6:00	2.2.2013 15:44	- 00.14:15:06
34	24.1.2013 6:00	23.1.2013 12:47	- 00.17:13:00	92	29.1.2013 6:00	28.1.2013 18:21	- 00.11:39:00
35	20.1.2013 6:00	19.1.2013 6:13	- 00.23:47:00	93	7.2.2013 6:00	11.2.2013 11:23	04.05:23:00
36	25.1.2013 6:00	28.1.2013 6:29	03.00:29:30	94	7.2.2013 6:00	10.2.2013 17:45	03.11:45:00
37	31.1.2013 6:00	4.2.2013 5:15	03.23:15:00	95	8.2.2013 6:00	7.2.2013 12:23	- 00.17:36:14
38	1.2.2013 6:00	28.1.2013 15:41	- 03.14:18:30	96	8.2.2013 6:00	7.2.2013 15:18	- 00.14:41:14
39	24.1.2013 6:00	23.1.2013 6:10	- 00.23:50:00	97	8.2.2013 6:00	7.2.2013 16:15	- 00.13:44:38
40	24.1.2013 6:00	23.1.2013 6:50	- 00.23:10:00	98	3.2.2013 6:00	1.2.2013 13:17	- 01.16:42:48
41	10.2.2013 6:00	12.2.2013 20:45	02.14:45:00	99	31.1.2013 6:00	21.2.2013 10:46	21.04:46:00
42	1.2.2013 6:00	31.1.2013 11:41	- 00.18:19:00	100	1.2.2013 6:00	31.1.2013 10:10	- 00.19:50:00
43	31.1.2013 6:00	3.2.2013 6:49	03.00:49:30	101	31.1.2013 6:00	30.1.2013 14:06	- 00.15:54:00
44	12.2.2013 6:00	13.2.2013 5:05	00.23:05:00	102	1.2.2013 6:00	31.1.2013 6:12	- 00.23:48:00
45	29.1.2013 6:00	28.1.2013 5:20	- 01.00:39:30	103	26.1.2013 6:00	25.1.2013 6:16	- 00.23:44:00
46	28.1.2013 6:00	27.1.2013 10:22	- 00.19:37:30	104	27.1.2013 6:00	26.1.2013 6:16	- 00.23:44:00
47	24.1.2013 6:00	24.1.2013 10:45	00.04:45:00	105	27.1.2013 6:00	26.1.2013 6:37	- 00.23:23:00
48	1.2.2013 6:00	31.1.2013 6:01	- 00.23:59:00	106	27.1.2013 6:00	26.1.2013 9:30	- 00.20:30:00
49	24.1.2013 6:00	23.1.2013 7:40	- 00.22:20:00	107	2.2.2013 6:00	1.2.2013 21:15	- 00.08:45:00
50	1.2.2013 6:00	1.2.2013 11:17	00.05:17:12	108	28.1.2013 6:00	27.1.2013 20:30	- 00.09:30:00
51	15.2.2013 6:00	22.2.2013 18:06	07.12:06:30	109	9.2.2013 6:00	8.2.2013 12:18	- 00.17:41:12
52	28.1.2013 6:00	28.1.2013 6:35	00.00:35:00	110	30.1.2013 6:00	1.2.2013 12:35	02.06:35:00
53	15.2.2013 6:00	20.2.2013 3:19	04.21:19:48	111	30.1.2013 6:00	1.2.2013 19:15	02.13:15:00
54	22.2.2013 6:00	27.2.2013 9:06	05.03:06:00	112	30.1.2013 6:00	29.1.2013 22:20	- 00.07:40:00
55	1.2.2013 6:00	2.2.2013 9:11	01.03:11:48	113	30.1.2013 6:00	29.1.2013 15:40	- 00.14:20:00
56	2.2.2013 6:00	6.2.2013 15:49	04.09:49:00	114	31.1.2013 6:00	29.1.2013 19:00	- 01.11:00:00
57	28.1.2013 6:00	27.1.2013 8:05	- 00.21:55:00	115	31.1.2013 6:00	1.2.2013 15:55	01.09:55:00
58	1.2.2013 6:00	1.2.2013 19:22	00.13:22:30	116	31.1.2013 6:00	30.1.2013 1:40	- 01.04:20:00

Příloha č. 5b: Plnění zákaznických termínů plného modelu MZ Liberec a.s.

Číslo zakázky	Termín stanovený zákazníkem	Termín reálného odvedení	Rozdíl mezi rozvrhovaným termínem a termínem zákazníka [dny.hodiny:minuty:sekundy]	Číslo zakázky	Termín stanovený zákazníkem	Termín reálného odvedení	Rozdíl mezi rozvrhovaným termínem a termínem zákazníka [dny.hodiny:minuty:sekundy]
117	31.1.2013 6:00	1.2.2013 22:35	01.16:35:00	175	31.1.2013 6:00	4.2.2013 20:26	04.14:26:00
118	31.1.2013 6:00	30.1.2013 7:45	- 00.22:15:00	176	31.1.2013 6:00	1.2.2013 19:41	01.13:41:00
119	27.1.2013 6:00	27.1.2013 5:15	- 00.00:45:00	177	11.2.2013 6:00	12.2.2013 6:53	01.00:53:00
120	26.1.2013 6:00	27.1.2013 19:56	01.13:56:00	178	30.1.2013 6:00	29.1.2013 8:34	- 00.21:26:00
121	22.1.2013 6:00	23.1.2013 17:24	01.11:24:00	179	31.1.2013 6:00	3.2.2013 8:56	03.02:56:30
122	29.1.2013 6:00	3.2.2013 9:06	05.03:06:00	180	26.1.2013 6:00	25.1.2013 17:25	- 00.12:35:00
123	22.1.2013 6:00	21.1.2013 16:21	- 00.13:39:00	181	31.1.2013 6:00	30.1.2013 15:14	- 00.14:45:24
124	9.2.2013 6:00	12.2.2013 9:05	03.03:05:00	182	7.2.2013 6:00	6.2.2013 17:10	- 00.12:50:00
125	28.1.2013 6:00	27.1.2013 16:58	- 00.13:01:12	183	20.2.2013 6:00	10.2.2013 12:33	- 09.17:27:00
126	29.1.2013 6:00	28.1.2013 14:52	- 00.15:07:48	184	4.2.2013 6:00	2.2.2013 18:21	- 01.11:39:00
127	27.1.2013 6:00	26.1.2013 10:28	- 00.19:32:00	185	23.1.2013 6:00	23.1.2013 6:30	00.00:30:00
128	31.1.2013 6:00	30.1.2013 16:33	- 00.13:27:00	186	31.1.2013 6:00	31.1.2013 14:21	00.08:21:06
129	1.2.2013 6:00	31.1.2013 10:58	- 00.19:02:00	187	15.2.2013 6:00	12.2.2013 20:10	- 02.09:50:00
130	25.1.2013 6:00	23.1.2013 14:49	- 01.15:11:00	188	24.1.2013 6:00	23.1.2013 18:50	- 00.11:09:30
131	29.1.2013 6:00	30.1.2013 5:25	00.23:25:00	189	5.2.2013 6:00	9.2.2013 6:50	04.00:50:00
132	25.1.2013 6:00	24.1.2013 18:40	- 00.11:20:00	190	30.1.2013 6:00	29.1.2013 18:34	- 00.11:26:00
133	27.1.2013 6:00	26.1.2013 13:02	- 00.16:58:00	191	24.1.2013 6:00	24.1.2013 4:25	- 00.01:35:00
134	24.1.2013 6:00	24.1.2013 7:05	00.01:05:00	192	24.1.2013 6:00	23.1.2013 6:54	- 00.23:06:00
135	25.1.2013 6:00	24.1.2013 1:40	- 01.04:20:00	193	27.1.2013 6:00	26.1.2013 16:25	- 00.13:34:42
136	3.2.2013 6:00	2.2.2013 17:48	- 00.12:11:36	194	31.1.2013 6:00	30.1.2013 14:25	- 00.15:34:24
137	31.1.2013 6:00	4.2.2013 12:45	04.06:45:00	195	24.1.2013 6:00	29.1.2013 12:55	05.06:55:00
138	7.2.2013 6:00	8.2.2013 20:40	01.14:40:24	196	31.1.2013 6:00	30.1.2013 6:54	- 00.23:06:00
139	28.1.2013 6:00	27.1.2013 12:50	- 00.17:09:30	197	20.1.2013 6:00	19.1.2013 16:00	- 00.14:00:00
140	6.2.2013 6:00	5.2.2013 22:47	- 00.07:12:30	198	24.1.2013 6:00	23.1.2013 12:30	- 00.17:30:00
141	27.1.2013 6:00	26.1.2013 6:01	- 00.23:58:48	199	3.2.2013 6:00	30.1.2013 2:27	- 04.03:32:48
142	27.1.2013 6:00	26.1.2013 7:51	- 00.22:09:00	200	26.1.2013 6:00	25.1.2013 20:46	- 00.09:14:00
143	1.2.2013 6:00	31.1.2013 19:22	- 00.10:38:00	201	29.1.2013 6:00	29.1.2013 2:00	- 00.04:00:00
144	27.1.2013 6:00	26.1.2013 6:25	- 00.23:34:36	202	26.1.2013 6:00	25.1.2013 8:09	- 00.01:51:00
145	31.1.2013 6:00	28.1.2013 9:55	- 02.20:04:30	203	25.1.2013 6:00	25.1.2013 3:05	- 00.02:55:00
146	4.2.2013 6:00	3.2.2013 2:01	- 01.03:58:30	204	25.1.2013 6:00	25.1.2013 1:25	- 00.04:35:00
147	31.1.2013 6:00	30.1.2013 6:02	- 00.23:58:00	205	29.1.2013 6:00	28.1.2013 14:27	- 00.15:33:00
148	9.2.2013 6:00	8.2.2013 7:45	- 00.22:15:00	206	25.1.2013 6:00	24.1.2013 22:15	- 00.07:45:00
149	27.1.2013 6:00	26.1.2013 6:16	- 00.23:44:00	207	10.2.2013 6:00	13.2.2013 18:39	03.12:39:48
150	27.1.2013 6:00	26.1.2013 6:29	- 00.23:31:00	208	20.1.2013 6:00	19.1.2013 14:55	- 00.15:04:12
151	26.1.2013 6:00	25.1.2013 6:03	- 00.23:57:00	209	1.2.2013 6:00	31.1.2013 6:02	- 00.23:58:00
152	24.1.2013 6:00	23.1.2013 12:08	- 00.17:52:00	210	30.1.2013 6:00	29.1.2013 6:20	- 00.23:40:00
153	24.1.2013 6:00	23.1.2013 12:21	- 00.17:39:00	211	11.2.2013 6:00	11.2.2013 16:08	00.10:08:00
154	6.2.2013 6:00	6.2.2013 14:25	00.08:25:00	212	27.1.2013 6:00	26.1.2013 15:19	- 00.14:40:12
155	10.2.2013 6:00	10.2.2013 12:50	00.06:50:00	213	27.1.2013 6:00	26.1.2013 12:16	- 00.17:43:48
156	28.1.2013 6:00	27.1.2013 15:20	- 00.14:40:00	214	20.1.2013 6:00	19.1.2013 12:50	- 00.17:09:42
157	26.1.2013 6:00	25.1.2013 15:20	- 00.14:40:00	215	7.2.2013 6:00	10.2.2013 14:19	03.08:19:36
158	27.1.2013 6:00	27.1.2013 5:35	- 00.00:25:00	216	31.1.2013 6:00	3.2.2013 9:11	03.03:11:00
159	3.2.2013 6:00	2.2.2013 18:51	- 00.11:09:00	217	29.1.2013 6:00	27.1.2013 19:42	- 01.10:17:48
160	24.1.2013 6:00	23.1.2013 12:05	- 00.17:55:00	218	31.1.2013 6:00	1.2.2013 18:06	01.12:06:48
161	24.1.2013 6:00	23.1.2013 17:50	- 00.12:10:00	219	12.2.2013 6:00	3.2.2013 21:26	- 08.08:34:00
162	27.1.2013 6:00	27.1.2013 8:02	00.02:02:00	220	31.1.2013 6:00	5.2.2013 3:45	04.21:45:00
163	30.1.2013 6:00	29.1.2013 18:18	- 00.11:41:39	221	7.2.2013 6:00	17.2.2013 14:32	10.08:32:00
164	27.1.2013 6:00	26.1.2013 18:08	- 00.11:51:39	222	5.2.2013 6:00	12.2.2013 1:04	06.19:04:00
165	31.1.2013 6:00	30.1.2013 8:24	- 00.21:36:00	223	30.1.2013 6:00	28.1.2013 9:08	- 01.20:51:18
166	20.1.2013 6:00	22.1.2013 9:10	02.03:10:00	224	27.1.2013 6:00	27.1.2013 0:08	- 00.05:51:30
167	20.1.2013 6:00	25.1.2013 4:40	04.22:40:00	225	5.2.2013 6:00	5.2.2013 14:07	00.08:07:48
168	31.1.2013 6:00	29.1.2013 13:34	- 01.16:26:00	226	1.2.2013 6:00	5.2.2013 6:29	04.00:29:30
169	31.1.2013 6:00	6.2.2013 9:19	06.03:19:30	227	28.1.2013 6:00	28.1.2013 7:00	00.01:00:30
170	7.2.2013 6:00	17.2.2013 15:43	10.09:43:42	228	9.2.2013 6:00	11.2.2013 6:05	02.00:05:48
171	1.2.2013 6:00	5.2.2013 0:50	03.18:50:00	229	7.2.2013 6:00	17.2.2013 1:57	09.19:57:30
172	9.2.2013 6:00	14.2.2013 16:11	05.10:11:00				
173	6.2.2013 6:00	7.2.2013 7:53	01.01:53:00				
174	31.1.2013 6:00	6.2.2013 7:11	06.01:11:48				

Příloha č. 6a: Využití pracovišť vybraného řešení plného modelu MZ Liberec a.s.

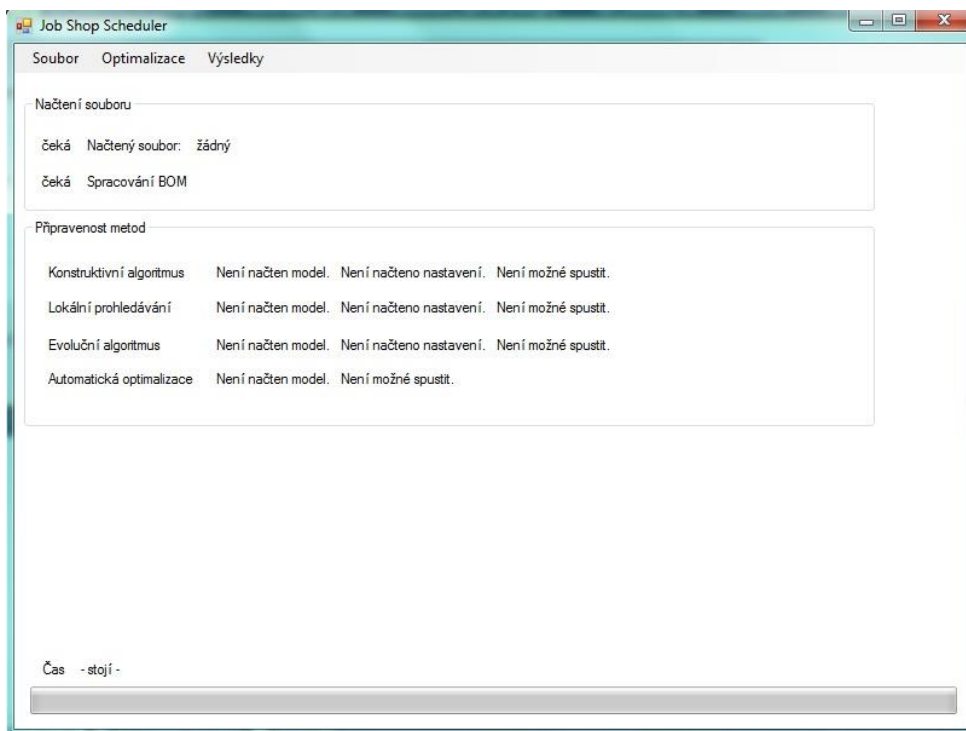
Stroj	Stroj je ve směně	Stroj je mimo směnu	Celkem - pracuje	Ve směně pracuje	Celkem - seřizování	Ve směně - seřizování	Celkem - nepracuje	Ve směně - nepracuje	Ve směně - práce je blokována	Mimo směnu - práce je blokována	Ve směně - seřizování je blokováno	Mimo směnu - seřizování je blokováno
0	63%	37%	1%	1%	0%	0%	62%	98%	0%	0%	2%	5%
1	63%	37%	4%	6%	1%	1%	58%	93%	2%	5%	1%	3%
2	63%	37%	4%	7%	0%	1%	58%	93%	2%	5%	5%	12%
3	63%	37%	1%	1%	0%	1%	61%	98%	0%	0%	2%	5%
4	63%	37%	1%	2%	0%	1%	61%	98%	1%	2%	2%	5%
5	63%	37%	5%	8%	0%	0%	57%	92%	4%	10%	3%	7%
6	63%	37%	5%	7%	0%	0%	58%	92%	4%	10%	2%	5%
7	63%	37%	5%	8%	0%	0%	58%	92%	2%	5%	3%	7%
8	63%	37%	22%	35%	2%	2%	39%	62%	12%	33%	3%	8%
9	63%	37%	30%	48%	1%	2%	32%	50%	13%	36%	6%	17%
10	63%	37%	9%	14%	1%	1%	53%	85%	5%	13%	3%	7%
11	63%	37%	0%	0%	0%	0%	62%	100%	0%	0%	2%	5%
12	63%	37%	1%	2%	0%	0%	61%	98%	0%	0%	2%	5%
13	63%	37%	0%	0%	0%	0%	63%	100%	0%	0%	0%	0%
14	63%	37%	0%	0%	0%	0%	62%	99%	0%	0%	0%	0%
15	63%	37%	12%	19%	1%	1%	50%	80%	5%	13%	2%	5%
16	63%	37%	7%	12%	0%	1%	55%	88%	4%	10%	1%	2%
17	63%	37%	6%	10%	1%	1%	56%	89%	1%	2%	9%	25%
18	63%	37%	12%	19%	1%	1%	50%	80%	7%	19%	7%	19%
19	63%	37%	16%	25%	1%	2%	46%	73%	8%	22%	7%	19%
20	63%	37%	25%	40%	1%	1%	37%	59%	18%	47%	1%	2%
21	63%	37%	17%	28%	1%	1%	45%	71%	14%	38%	0%	0%
22	63%	37%	16%	26%	0%	1%	46%	74%	11%	29%	3%	7%
23	63%	37%	14%	22%	1%	1%	48%	77%	12%	33%	0%	0%
24	63%	37%	18%	29%	1%	1%	44%	71%	11%	30%	2%	5%
25	63%	37%	15%	24%	0%	0%	47%	76%	9%	23%	2%	5%
26	63%	37%	26%	42%	0%	1%	36%	57%	15%	41%	0%	0%
27	63%	37%	6%	10%	1%	1%	56%	89%	5%	12%	12%	32%
28	63%	37%	41%	66%	10%	15%	12%	19%	25%	66%	7%	18%
29	63%	37%	8%	13%	0%	0%	54%	86%	5%	15%	2%	5%
30	63%	37%	1%	1%	0%	0%	62%	99%	0%	0%	1%	2%
31	63%	37%	4%	6%	0%	0%	59%	94%	4%	9%	2%	5%
32	63%	37%	2%	3%	0%	0%	61%	97%	0%	0%	0%	0%
33	63%	37%	3%	6%	0%	0%	59%	94%	4%	10%	0%	0%
34	63%	37%	0%	0%	0%	0%	63%	100%	0%	0%	2%	5%
35	63%	37%	8%	12%	0%	0%	55%	87%	7%	19%	2%	5%
36	63%	37%	7%	11%	0%	0%	55%	89%	7%	19%	1%	2%
37	63%	37%	19%	30%	0%	0%	44%	70%	12%	33%	1%	2%
38	63%	37%	7%	11%	0%	0%	55%	89%	7%	19%	1%	2%
39	63%	37%	5%	7%	0%	0%	58%	93%	3%	8%	0%	0%
40	63%	37%	0%	0%	0%	0%	62%	100%	1%	3%	0%	0%
41	63%	37%	11%	17%	0%	0%	52%	82%	7%	17%	0%	0%
42	63%	37%	5%	8%	0%	0%	57%	92%	5%	14%	0%	0%
43	63%	37%	0%	0%	0%	0%	63%	100%	0%	0%	1%	2%
44	63%	37%	2%	3%	0%	0%	61%	97%	0%	0%	2%	5%
45	63%	37%	9%	14%	0%	1%	54%	86%	7%	20%	1%	3%
46	63%	37%	0%	0%	0%	0%	62%	100%	0%	0%	2%	5%
47	63%	37%	0%	0%	0%	0%	62%	100%	0%	0%	0%	0%
48	63%	37%	12%	19%	0%	1%	51%	81%	5%	13%	5%	15%
49	63%	37%	3%	5%	0%	0%	60%	95%	3%	7%	0%	0%
50	63%	37%	3%	5%	0%	0%	60%	95%	2%	5%	0%	0%

Příloha č. 6b: Využití pracovišť vybraného řešení plného modelu MZ Liberec a.s.

Stroj	Stroj je ve směně	Stroj je mimo směnu	Celkem - pracuje	Ve směně pracuje	Celkem - seřizování	Ve směně - seřizování	Celkem - nepracuje	Ve směně - nepracuje	Ve směně - práce je blokována	Mimo směnu - práce je blokována	Ve směně - seřizování je blokováno	Mimo směnu - seřizování je blokováno
51	63%	37%	31%	50%	4%	6%	28%	44%	26%	70%	5%	15%
52	63%	37%	2%	3%	0%	0%	60%	96%	2%	5%	1%	2%
53	63%	37%	0%	0%	0%	0%	63%	100%	0%	0%	1%	2%
54	63%	37%	0%	0%	0%	0%	62%	100%	0%	0%	1%	2%
55	63%	37%	0%	1%	0%	0%	62%	99%	1%	2%	0%	0%
56	63%	37%	0%	0%	0%	0%	63%	100%	0%	0%	0%	0%
57	63%	37%	0%	0%	0%	0%	62%	100%	0%	0%	1%	2%
58	63%	37%	2%	3%	0%	0%	61%	97%	0%	0%	2%	5%
59	63%	37%	0%	0%	0%	0%	63%	100%	0%	0%	0%	0%
60	63%	37%	0%	0%	0%	0%	62%	100%	1%	2%	0%	0%
61	63%	37%	1%	2%	0%	0%	61%	98%	0%	0%	3%	7%
62	63%	37%	2%	3%	0%	0%	61%	97%	0%	0%	1%	2%
63	63%	37%	0%	0%	0%	0%	63%	100%	0%	0%	0%	0%
64	63%	37%	0%	0%	0%	0%	62%	100%	0%	0%	2%	5%
65	63%	37%	3%	5%	0%	1%	59%	94%	1%	3%	4%	10%
66	63%	37%	2%	3%	0%	0%	60%	96%	1%	3%	0%	0%
67	63%	37%	0%	0%	0%	0%	63%	100%	0%	0%	0%	0%
68	63%	37%	0%	0%	0%	0%	63%	100%	0%	0%	0%	0%
69	63%	37%	2%	4%	0%	0%	60%	96%	3%	7%	1%	2%
70	63%	37%	1%	1%	0%	0%	62%	99%	0%	0%	0%	0%
71	63%	37%	0%	0%	0%	0%	63%	100%	0%	0%	0%	0%
72	63%	37%	2%	3%	0%	0%	61%	97%	0%	0%	2%	5%
73	63%	37%	1%	2%	0%	0%	61%	98%	1%	3%	0%	0%
74	63%	37%	0%	0%	0%	0%	62%	100%	0%	0%	0%	0%
75	63%	37%	1%	2%	0%	0%	61%	98%	1%	3%	0%	0%
76	63%	37%	6%	10%	0%	0%	56%	90%	4%	10%	1%	2%
77	100%	0%	1%	1%	0%	0%	99%	99%	0%	0%	0%	0%
78	100%	0%	3%	3%	0%	0%	97%	97%	0%	0%	0%	0%
79	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
80	100%	0%	11%	11%	0%	0%	89%	89%	0%	0%	0%	0%
81	100%	0%	35%	35%	0%	0%	65%	65%	0%	0%	0%	0%
82	100%	0%	1%	1%	0%	0%	99%	99%	0%	0%	0%	0%
83	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
84	100%	0%	2%	2%	0%	0%	98%	98%	0%	0%	0%	0%
85	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
86	100%	0%	2%	2%	0%	0%	98%	98%	0%	0%	0%	0%
87	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
88	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
89	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
90	100%	0%	1%	1%	0%	0%	99%	99%	0%	0%	0%	0%
91	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
92	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
93	100%	0%	28%	28%	0%	0%	72%	72%	0%	0%	0%	0%
94	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
95	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
96	100%	0%	6%	6%	0%	0%	94%	94%	0%	0%	0%	0%
97	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
98	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
99	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
100	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%
101	100%	0%	0%	0%	0%	0%	100%	100%	0%	0%	0%	0%

Příloha č. 7: Rozvrhovací nástroj

Rozvrhovací nástroj je vyvíjen v prostředí Microsoft Visual Basic 2005 a 2010 express edition a to spíše jako testovací nástroj než plnohodnotný aplikovatelný SW v rámci dílenského řízení výroby. Přesto, že poskytuje všechna potřebná data pro realizaci dílenského rozvrhu, není předem určený pro plnou aplikaci ve výrobě. Má spíše demonstrovat možnosti aplikace pokročilých rozvrhovacích metod v některém z profesionálních APS či MES systémů (obr. P7-1).



Obr. P7-1: Hlavní obrazovka rozvrhovacího nástroje

Vstupní data

Vzhledem k předpokladu aplikování daných principů a SW pro SME podniky zabývající se zakázkovou výrobou, kde je malé množství operací s dlouhými procesními časy, je zdrojem všech vstupních dat o výrobním systému model vytvořený ve formátu xls. Při volbě formátu xls byl také brán v potaz fakt, že většina dílenských plánovačů již MS Excell využívá pro plánování a popř. rozvrhování, tudíž nebudou mít problémy při tvorbě modelu výrobního systému.

Vstupní data o výrobním systému obsahují (obr. P7-2):

- Technologický postup, včetně termínu uvolnění zakázky do výroby a požadovaném termínu odvedení zakázky.
- Údaje o omezeních (strojích) včetně parametrů týkajících se směnnosti.
- Transportní matice udávající časovou náročnost přesunu výrobku na další pracoviště.

1									
Zakázka	Počet kusů	Optimální dávka	Minimální dávka	Termín náběhu	Termín odvedení	Stroj	Kus. č.[min]	Dáv. č.[min]	Typ přes.
1	1	1	1	16.4.2012 6:00	18.4.2012 6:00	9	65	0	0
2	1	1	1	16.4.2012 6:00	18.4.2012 6:00	0	64	0	0
3	1	1	1	16.4.2012 6:00	18.4.2012 6:00	14	72	0	0
4	1	1	0	0	0	0	0	0	0
5	1	1	1	15	0	0	0	0	0
6	1	1	2	23	20	0	0	0	0
7	1	1	3	40	34	0	480	0	3
8	1	1	4	35	28	1	480	0	3
9	1	1	5	1440	45	2	480	0	3
10	1	1	6	80	78	3	480	0	3
11	1	1	7	75	664	4	480	0	3
12	1	1	8	60	523	5	480	0	3
13	1	1	9	45	25	6	480	0	3
14	1	1	10	50	25	7	480	0	3
15	1	1	11	45	45	8	480	0	3
			12	30	25	9	480	0	3
			13	72	48	10	480	0	3
			14	84	25	11	480	0	3
			Sklad	60	60	12	480	0	3
						13	480	0	3
						14	480	0	3
						15	480	0	3

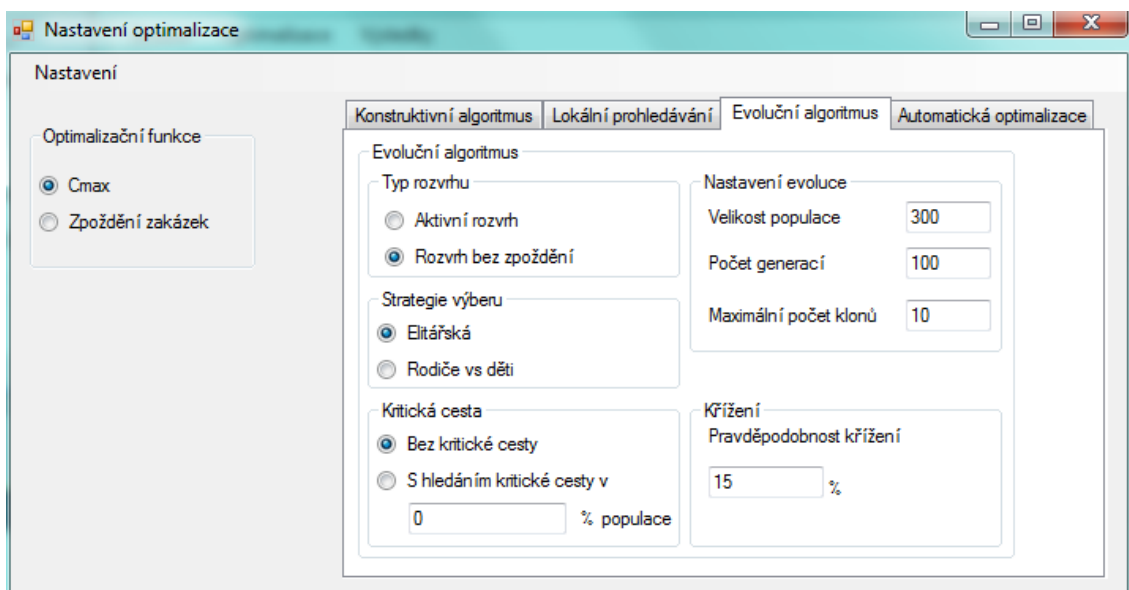
Obr. P7-2: Model výrobního systému

Nastavení

Nastavení optimalizačních parametrů obsahuje základní volbu účelové funkce a základních parametrů jednotlivých optimalizačních metod:

- Konstruktivní algoritmus
 - Druh rozvrhu – aktivní (A) a bez zpoždění (ND).
 - Prioritní pravidla - SPT, LPT, MKWR, FIFO a RS, u kterého je možné navrhnout počet vygenerovaných náhodných rozvrhů, V případě, že je zvolen jak aktivní rozvrh, tak i rozvrh bez zpoždění, stanovuje toto číslo počet všech náhodně generovaných rozvrhů (tzn. v případě 10 rozvrhů je 5 aktivních a 5 bez zpoždění).
- Lokální prohledávání – možnost optimalizovat všechna počáteční řešení generované pomocí CA, či pouze to nejlepší. Algoritmus sám rozpozná druh rozvrhu, kterým bylo počáteční řešení vygenerováno a optimalizuje ho dle původního druhu rozvrhu.
- Evoluční algoritmus (obr. P7-3)
 - Druh rozvrhu – A či ND.
 - Strategie výběru (selekce) rodičů pro tvorbu nových řešení – elitářský a turnajový princip.
 - Hybridizace lokálním prohledáváním – použití mutace lokálním prohledáváním popř. procento rodičů, na kterých bude provedena mutace.
 - Velikost populace - počet jedinců reprezentujících řešení chromosomem.
 - Počet generací – tzn. počet iterací EA.

- Maximální počet klonů - určení počtu možných stejných řešení v populaci v jedné iteraci.
- Pravděpodobnost křížení – pravděpodobnost, že bude sdílen každý gen z druhého rodiče.
- Automatická optimalizace - čas, za jak dlouho má být ukončena optimalizace:
 - Absolutní – zadání přesného data a času ukončení optimalizace
 - Relativní – zadání délky optimalizace



Obr. P7-3: Nastavení evolučního algoritmu

Toto nastavení je uchováváno v textovém formátu a lze je upravovat a ukládat v prostředí vyvíjeného SW (doporučeno), či pomocí textového editoru.

Výstupní data

Výstupní data jsou rozdělena do základních 2 oblastí a to na data potřebná pro realizaci dílenského rozvrhu a optimalizační statistiky jednotlivých metod, určené pro výzkum v oblasti těchto algoritmů.

Data potřebná pro realizaci dílenského rozvrhu pak jsou :

- Seznam operací pro dopředný a zpětný rozvrh (obr. P7-4):
 - identifikace operace se zakázkou
 - stroj, na kterém se operace provádí
 - procesní, dávkový a transportní čas
 - datum a čas předpokládaného začátku a konce operace

Gantt_form

LS - 6-LS_A-65 makespan - 55

Ukaž výsledky Měřitko ganttových grafů 1 Přenastav měřitko Ulož rozvrh do excelu

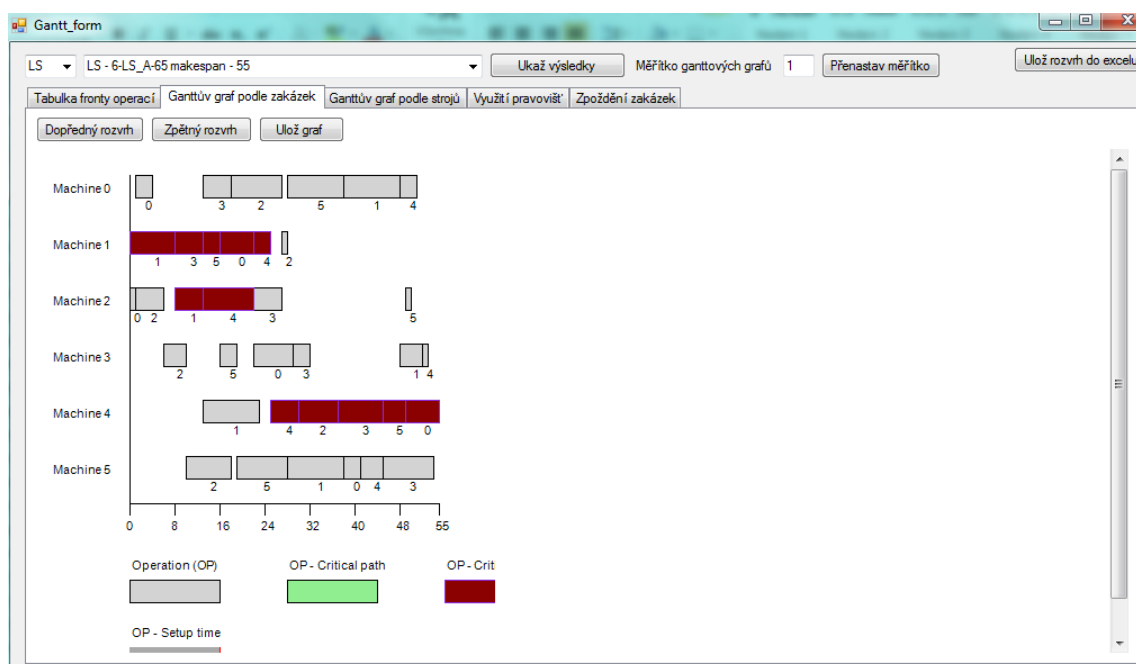
Tabulka fronty operací Ganttův graf podle zakázek Ganttův graf podle strojů Využití pravovištví Zpoždění zakázek

Dopředný rozvrh Zpětný rozvrh

	Číslo operace	Stroj	Začátek dávky	Konec dávky	Sekvence	Zbývající čas do konce zakázky	Zakázka	Procesní čas	Začátek dávky (datum)	Konec dávky (datum)	Seřizovací čas	Transport
▶	0	2	0	1	0	26	0	1	1.1.2009 22:00:00	1.1.2009 22:01:00	0	0
	1	0	1	4	2	25	0	3	1.1.2009 22:01:00	1.1.2009 22:04:00	0	0
	2	1	16	22	11	22	0	6	1.1.2009 22:16:00	1.1.2009 22:22:00	0	0
	3	3	22	29	19	16	0	7	1.1.2009 22:22:00	1.1.2009 22:29:00	0	0
	4	5	38	41	25	9	0	3	1.1.2009 22:38:00	1.1.2009 22:41:00	0	0
	5	4	49	55	35	6	0	6	1.1.2009 22:49:00	1.1.2009 22:55:00	0	0
	6	1	0	8	1	47	1	8	1.1.2009 22:00:00	1.1.2009 22:08:00	0	0

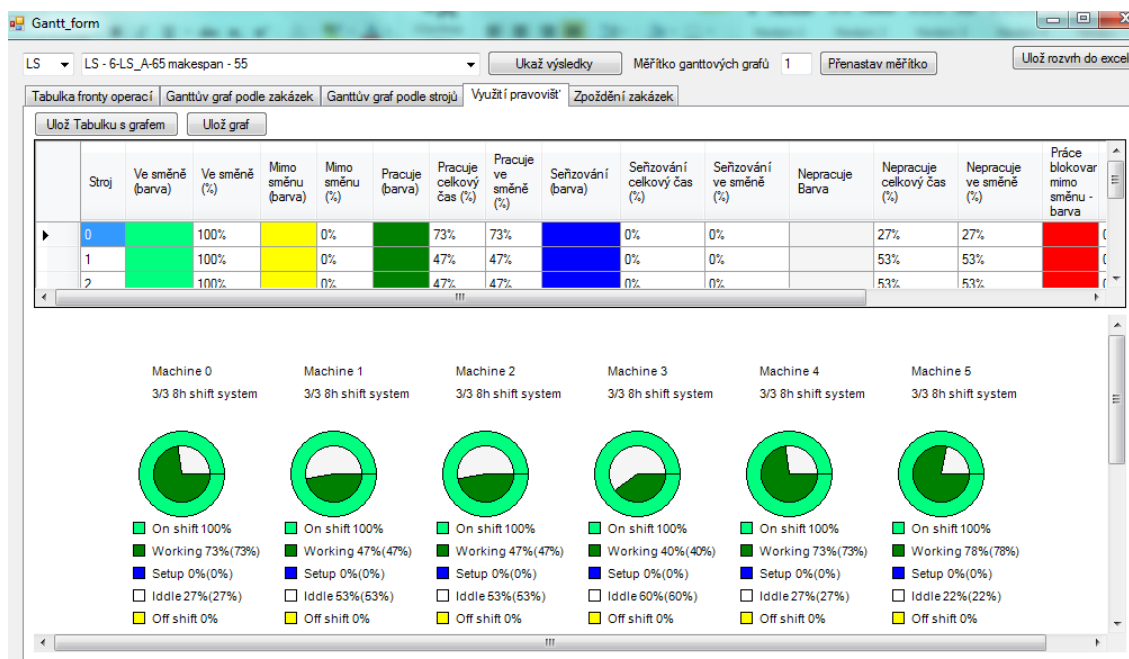
Obr. P7-4: Výpis rozvrhu operací

- Vizualizace rozvrhu pro Ganttův graf z pohledu stroje (obr.P7-5) či zakázky – obsahuje analýzu kritické cesty a kritických bloků. Je vhodné jej použít v případech ohrožení dílenského plánu nesplněním lhůt a při kontrole realizace dílenského rozvrhu v případě, že je rozvrženo malé množství operací. Při větším počtu operací lze nastavit měřitko zobrazení tohoto grafu, avšak rozvrh je ze své podstaty nepřehledný. V takových situacích je vhodnější použít výpis rozvrhu. Při zobrazení pohledu, kdy na ose y jsou zobrazeny stroje, je možné sledovat seřízení. V zobrazení, kdy na ose y jsou zakázky, lze sledovat transport.



Obr. P7-5: Ganttův graf rozvrhu

- Využití pracovišť (obr. P7-6):
 - využití dostupného času (dle směnnosti) a celkového času pro:
 - pracovní operace,
 - seřizovací operace.
 - blokace zakázky z důvodu práce následujícího stroje v technologickém postupu (pracovní operace a seřizování).



Obr. P7-6: Složené koláčové grafy využití pracovišť

Na základě těchto výstupů lze stanovit kritické pracoviště či přijmout opatření změny směnnosti pracoviště.

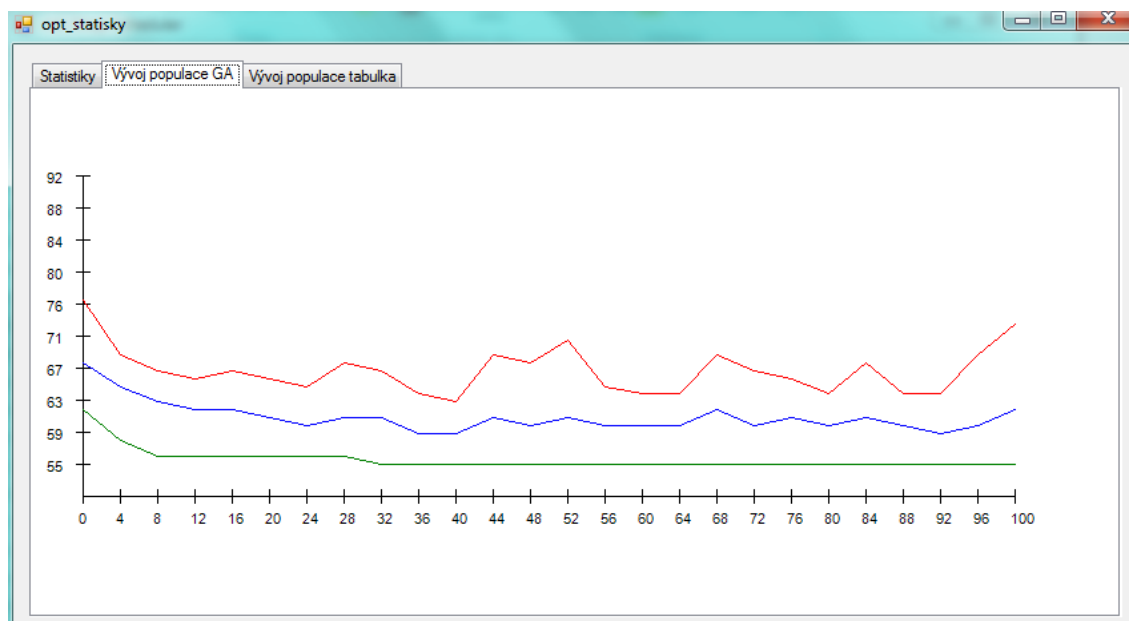
- Plnění termínu zakázek – vzhledem k možnosti dávkové výroby jsou jednotlivé zakázky v zobrazení výstupu řazeny dle výrobních dávek. To se při velkém množství zakázek a jejich dělení může stát nepřehledným, proto lze sledovat termín dokončení a zpoždění oproti zákaznickému termínu odděleně dle zakázek (obr. P7-7).

	Číslo zakázky	Termín odvedení zákazník	Termín reálného odvedení	Zpoždění D:H:M:S
▶	1	1.1.2009 22:55:00	1.1.2009 22:55:00	00:00:00:00
	2	1.1.2009 22:52:00	1.1.2009 22:52:00	00:00:00:00
	3	1.1.2009 22:37:00	1.1.2009 22:37:00	00:00:00:00
	4	1.1.2009 22:54:00	1.1.2009 22:54:00	00:00:00:00

Obr. P7-7: Sledování zákaznických termínů

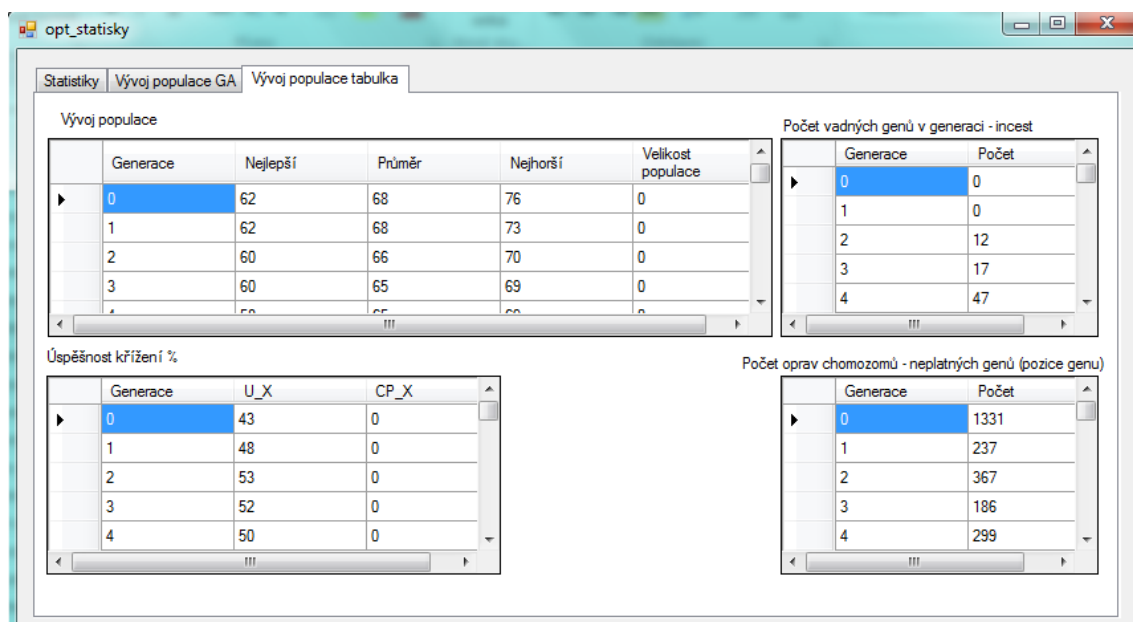
Optimalizační statistiky mimo hodnoty účelových funkcí jednotlivých metod poskytují údaje o průběhu evolučního algoritmu.

Pro sledování populačního vývoje (obr. P7-7), zejména konvergence populace k lokálnímu či globálnímu extrému je využitý graf znázorňující vývoj nejlepšího (zelená barva) jedince, průměrné hodnoty účelové funkce (modrá barva) v populaci a hodnota nejhoršího jedince (červená barva). Osa x popisuje n -tou iteraci, osa y pak hodnotu účelové funkce.



Obr. P7-8: Vývoj účelové funkce

Vývoj populace lze sledovat i díky výpisu předešlého grafu pro další zpracování (obr. P7-9). Optimalizační statistiky zobrazují úspěšnost daného křížení (uniformní – U_X a mutace pomocí hybridizace CP_X), tzn. procentuální podíl zlepšujících křížení k celkovému počtu operací křížení. Identifikace křížení, které zlepšuje účelovou funkci vychází z hodnoty účelové funkce rodiče a potomka.



Obr. P7-9: Optimalizační statistiky

Statistiky dále obsahují vývoj počtu opravených genů v rámci opravy poškozeného chromosomu a počet možných incestních křížení.